

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

AÑO II N.º 57

160 Ptas.

Canarias 165 pts.

**LOTERIA PRIMITIVA
EN LOGO****PLANIFICACION
DE
PROYECTOS:
EL CAMINO
MAS CORTO****SONIMAG 86...
E IBERIA****GESTION DE
FICHEROS INTELIGENTE****NUMEROS DE
IMPRESION EN
LENGUAJE
MAQUINA****A tiro limpio
con West Bank**

LAS TRES LUCES

LAS TRES LUCES DE GLAURUNG

**UN PROGRAMA
H E C H O
EN ESPAÑA
QUE ESTA
SORPRENDIENDO
EN EUROPA**

Un guerrero va a enfrentarse, sólo, a los incontables peligros que acechan en el Castillo bajo la Montaña, más allá de donde alcanza la luz del Sol y de donde se atreven a llegar los corazones más valerosos.

La fuerza, la astucia y la habilidad, van a medirse con el hierro, el fuego y la hechicería, en uno de los más tremendos choques entre el Bien y el Mal que el Universo ha presenciado jamás.

ERBE

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA
ERBE SOFTWARE,
SANTA ENGRACIA, 17. Tel. 447 34 10
DELGACION BARCELONA,
Avd. MISTRAL, 10. Tel. (93) 432 07 31

Tarjeta de suscripción ahorro

■ **Deseo suscribirme a la revista Microhobby Amstrad** durante un año (50 números) por sólo 6.900 ptas., lo que me supone un ahorro de 1.100 ptas. Además, quedará suscrito automáticamente, y sin ningún cargo adicional, a Amstrad Cassette por seis meses.

■ **Deseo suscribirme a las cintas de Microhobby Amstrad** (12 cintas), al precio de 8.175 ptas., lo que me supone un ahorro de 900 ptas. La primera cinta que deseo recibir es la _____

NOMBRE _____ Fecha de nacimiento _____

APELLIDOS _____

DOMICILIO _____

CIUDAD _____ PROVINCIA _____

C. POSTAL _____ TELEFONO _____ PROFESION _____

Marco con una (x) en el casillero correspondiente la forma de pago que más conviene. Fecha y firma _____

☐ Tarjeta de crédito n.º _____

☐ Visa ☐ American Express ☐ Master Charge

Fecha de caducidad de la tarjeta _____

☐ Talón bancario a nombre de HOBBY PRESS, S. A.

☐ Giro Postal a nombre de HOBBY PRESS, S. A., N.º del giro _____

☐ Contra reembolso del primer envío. (Supone 75 ptas. de gastos de envío).

Si lo prefiere, también puede suscribirse por teléfono.

734 65 00



28

erros
eresante rutina inteligente para
ficheros. Está llena de trucos
render, pero, naturalmente,

Consejero Delegado
José I. Gómez-Centurión

Jefe de Producción
Carlos Peropadre

Marketing
Marta García

Jefe de Publicidad
Concha Gutiérrez

Secretaría de Dirección
Pilar Arestizábal

Suscripciones
M.º Rosa González
M.º del Mar Calzada

Redacción, Administración y Publicidad

Ctra. de Irún km 12,400
(Fuencarral) 28049 Madrid

Pedidos y suscripciones:
734 65 00

Redacción: 734 70 12

Dto. Circulación
Paulino Blanco

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
ROTEDEC, S. A. Crta. de Irún.
Km. 12,450 (MADRID)

Fotocomposición
Novocomp, S.A.
Nicolás Morales, 38-40

Fotomecánica
GROF
Ezequiel Solana, 16

Depósito Legal:
M-28468-1985

Derechos exclusivos
de la revista

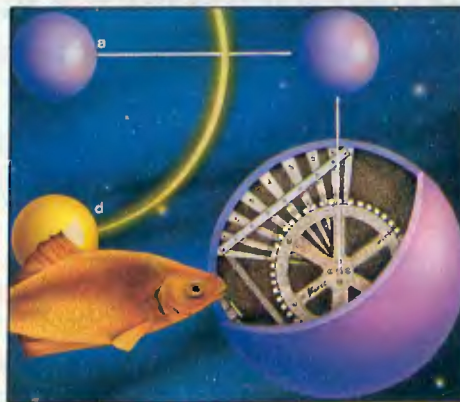
**COMPUTING with
the AMSTRAD**

Representante para Argentina, Chile,
Uruguay y Paraguay, Cia.
Americana de Ediciones, S.R.L. Sud
América 1.532. Tel.: 21 24 64. 1209
BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace
necesariamente solidario de las
opiniones vertidas por sus
colaboradores en los artículos
firmados. Reservados todos los
derechos.

Gestión

La planificación de proyectos es una de las áreas de la gestión en la que los ordenadores, debido a su gran capacidad de cálculo, pueden ser más útiles. El programa «Camino crítico», que acompaña al artículo, así lo demuestra.



32

Código Máquina

La impresión de números en pantalla, en hexadecimal y decimal, puede parecer un asunto bastante trivial, y, visto desde Basic, sin duda lo es. Pero cuando se trata de lenguaje máquina, la cosa cambia radicalmente; no es que se vuelva incomprensible, pero hay unos cuantos puntos a aclarar, precisamente los que se tratan en este artículo.

Mr. Joystick 26

West Bank es una divertida aventura en el Oeste Americano, en la que debemos salvar a unos cuantos inocentes a tiro limpio. Diversión y adicción garantizadas.



LUGAR
DESTINADO
PARA
EL SELLO

HOBBY PRESS, S. A.

Apartado de Correos N.º 232
ALCOBENDAS
(Madrid)



los incontables
peligros que
acechan en el
Castillo bajo la
Montaña, más allá
de donde alcanza la
luz del Sol y de
donde se atreven a
llegar los corazones
más valerosos.

La fuerza, la astucia
y la habilidad, van a
medirse con el
hierro, el fuego y la
hechicería, en uno
de los más
tremendos choques
entre el Bien y el
Mal que el Universo
ha presenciado
jamás.

ERBE

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA
ERBE SOFTWARE.
SANTA ENGRACIA, 17. Tel. 447 34 10
DELEGACION BARCELONA.
Avd. MISTRAL, 10. Tel. (93) 432 07 31

MICROHOBBY

AMSTRAD

Sumario

Director Editorial
José I. Gómez-Centurión

Director Ejecutivo
José M.º Díaz

Redactor Jefe
Juan José Martínez

Diseño gráfico
Fernando Chaumel

Colaboradores

Eduardo Ruiz
Javier Barceló
David Sopena
Robert Chatwin
Francisco Portalo
Pedro Sudón
Miguel Sepúlveda
Francisco Martín
Jesús Alonso
Pedro S. Pérez
Amalio Gómez
Alberto Suñer

Secretaría Redacción
Carmen Santamaría

Fotografía
Carlos Candel
Chema Sacristán

Portada
M. Barco

Ilustradores
J. Igual, J. Pons, F. L. Frontán,
J. Septien, Pejo, J. J. Mora

Edita
HOBBY PRESS, S.A.

Presidente
María Andriño

Consejero Delegado
José I. Gómez-Centurión

Jefe de Producción
Carlos Peropadre

Marketing
Marta García

Jefe de Publicidad
Concha Gutiérrez

Secretaría de Dirección
Pilar Arestizábal

Suscripciones
M.º Rosa González
M.º del Mar Calzado

Redacción, Administración y Publicidad

Ctra. de Irún km 12,400
(Fuencarral) 28049 Madrid

Pedidos y suscripciones:
734 65 00

Redacción: 734 70 12

Dto. Circulación
Paulino Blanco

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
ROTECIC, S. A. Ctra. de Irún,
Km. 12,450 (MADRID)

Fotocomposición
Novacom, S.A.
Nicolás Morales, 38-40

Fotomecánica
GROF
Ezequiel Solano, 16

Depósito Legal:
M-28468-1985

Derechos exclusivos
de la revista

**COMPUTING with
the AMSTRAD**

Representante para Argentina, Chile,
Uruguay y Paraguay, Cia.
Americana de Ediciones, S.R.L. Sud
América 1.532. Tel.: 21 24 64. 1209
BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace
necesariamente solidaria de las
opiniones vertidas por sus
colaboradores en los artículos
firmados. Reservados todos los
derechos.

Año II • Número 57 • 14 al 20 de octubre de 1986
160 ptas. (incluido I.V.A.)
Canarias, 155 ptas. + 10 ptas. sobretasa aérea.
Ceuta y Melilla, 155 ptas.

Primera plana 5

Novedades **Amstrad:**
Dinamic lanza el «Cobra Arc».

16 Gestión

La planificación de proyectos es una de las áreas de la gestión en la que los ordenadores, debido a su gran capacidad de cálculo, pueden ser más útiles. El programa «Camino crítico», que acompaña al artículo, así lo demuestra.



Mr. Joystick 26

West Bank es una divertida aventura en el Oeste Americano, en la que debemos salvar a unos cuantos inocentes a tiro limpio. Diversión y adicción garantizadas.



Gestión ficheros 28

Os ofrecemos una interesante rutina inteligente para la gestión de carga de ficheros. Está llena de trucos que merece la pena aprender, pero, naturalmente, sólo es útil para el disco.



Código Máquina 32

La impresión de números en pantalla, en hexadecimal y decimal, puede parecer un asunto bastante trivial, y, visto desde Basic, sin duda lo es. Pero cuando se trata de lenguaje máquina, la cosa cambia radicalmente; no es que se vuelva incomprensible, pero hay unos cuantos puntos a adarar, precisamente los que se tratan en este artículo.



SOMOS MAYORISTAS

MICRO-1

PRECIOS
INCLUIDO IVA

C/ Duque de Sesto, 50. 28009 Madrid
Tel.: (91) 275 96 16 - 274 75 02

(Metro O'Donnell o Goya)
Aparcamiento gratuito en Felipe II

OFERTA DE SOFTWARE: 2 PROGRAMAS AL PRECIO DE 1

JACK THE NIPPER	2.100 ptas.	KNIGHT GAMES	2.300 ptas.
BAT MAN	2.300 ptas.	STAINLESS STEEL	2.300 ptas.
ROCK'N LUCHA	1.395 ptas.	SABRE WULF	1.695 ptas.
THE WAY OF THE TIGER	2.300 ptas.	SABOTEUR	2.300 ptas.
WEST BANK	1.395 ptas.	PING PONG	2.300 ptas.
CAMELOT WARRIORS	2.300 ptas.	EQUINOX	2.300 ptas.
WORLD CUP	1.395 ptas.	CAULDRON II	2.300 ptas.
DECATHLON	495 ptas.	BEACH HEAD	495 ptas.
DRAGONTORC	495 ptas.	SOUTHERN BELLE	495 ptas.
WORLD CUP (DISCO)	3.300 ptas.	MOVIE (DISCO)	3.300 ptas.
GREEN BERET (DISCO)	3.300 ptas.		

SOFTWARE DE REGALO (OFERTA 2 x 1)
DECATHLON BEACH HEAD SOUTHERN BELLE
DRAGONTORC

LAPIZ OPTICO
3.295 PTAS.

CASSETTE ESPECIAL ORDENADOR
4.495 PTAS.

SINTETIZADOR DE VOZ EN
CASTELLANO
7.650 PTAS.

AMPLIACION DE MEMORIA ANTA 64 K.3
12.500 PTAS.

IMPRESORAS 20 % DE DESCUENTO SOBRE P.V.P.

TAPA DE METACRILATO 464	895
CABLE CENTRONICS	3.175
CABLE SEGUNDA UNIDAD D.	1.790
CABLE SEPARADORES 6128	1.975
INTERFACE RS232	9.265
CINTA VIRGEN C15	69

CABLE AUDIO	795
CABLE ADAPTADOR 2 JOYSTICK	2.390
CABLE SEPARADORES 464	1.390
CABLE SEPARADOR 8256	2.900
CABLE RS232	2.500
DISKETTES 3"	990

COMPATIBLE PC-IBM
SOFTWARE DE GESTION
2 BOCAS DE DISCO DE 360 K
MONITOR Y TECLADO EN CASTELLANO
169.000 PTAS.

CONVERTIDOR DE MONITOR AMSTRAD
EN TELEVISION
21.900 PTAS.

PRECIOS EXCEPCIONALES PARA TU AMSTRAD CPC-464, CPC-6128, PCW-8256, PCW-8512

OFERTAS EN JOYTICKS

QUICK SHOT I	1.395
QUICK SHOT II	1.695
QUICK SHOT IX	2.395

RATON DE PANTALLA
CON SOFTWARE 8.900 PTAS.

PEDIDOS CONTRA REEMBOLSO SIN NINGUN GASTO DE ENVIO. TEL. (91) 275 96 16 - 274 75 02
O ESCRIBIENDO A: MICRO-1. C/ DUQUE DE SESTO, 50. 28009 MADRID.

Tiendas y distribuidores grandes descuentos.
Dirigirse a Diproimsa. C/ Galatea, 25. Tel. (91) 274 75 03

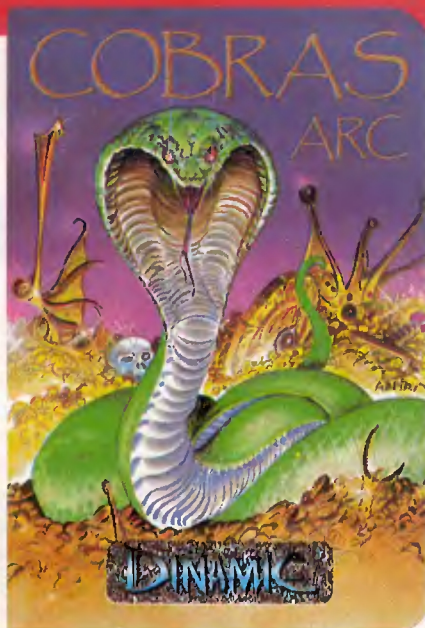
COBRA ARC DE DINAMIC

Los sacerdotes de una cultura milenaria vivían plácidamente en un reino lejano, dedicados por entero a la veneración del Dios Cobra.

En lugar construyeron un templo inmenso en el que durante siglos fueron depositando: tesoros, joyas, oro y toda una suerte de riquezas hasta que la tradición popular hizo suya la leyenda y las generaciones pasaban en vida pensando dónde se encontraría el tesoro del Templo Cobra.

En realidad los sacerdotes para proteger toda esa inmensa riqueza habían construido alrededor del tesoro un inmenso barro tan grande como una pirámide, asustados por las profecías del adivino Werdhal que vaticinaba un maremoto y una galerna que asolaría sus tierras.

La profecía se cumplió y el tesoro del templo quedó perdido entre las



gigantescas olas que lo transportaron lejos, muy lejos; nadie sabía dónde...

EL PRESENTE:

Nuevas generaciones de aventureros han leído los viejos pergaminos donde se habla del tesoro y del Templo Cobra.

Todos quieren encontrarlo, pero pocos saben cómo llegar hasta él.

Puedes partir de tres ciudades diferentes:

1. Hyrga: Cordillera del Himalaya al Suroeste del templo Cobra.

Primera PLANA

2. Greystone: Ciudad amurallada toda ella de piedra y metal.

3. Vendha: Ciudad del terrible, desierto de Niobia, es difícil sobrevivir en él.

CLAVES ARGUMENTALES:

El objetivo del juego consiste en llegar al Templo Cobra y rescatar el tesoro.

Para lo cual tienes que efectuar unos pasos obligatorios:

1. Matar al Príncipe Cobra.

2. Matar al hechizero.

— Cuando has acabado con los dos, estás en situación de llegar al templo Cobra.

— Para entrar en él necesitas la llave, y para conseguirla debes darle un regalo a un personaje determinado.

— Procura permanecer el menor tiempo posible en el mar porque se desatan muchas tormentas peligrosas.

— Para acabar con los dragones, necesitas utilizar la espada.

AMSTRAD COMPUTER SHOW

Bajo una impecable organización, así como un férreo y casi invisible sistema de seguridad, se ha celebrado en Londres, durante los días 3, 4 y 5 de octubre, el **Amstrad Computer Show**.

La edición de este año ha estado marcada por una predilección muy especial hacia las comunicaciones. Más de 10 modelos distintos de modems se presentaban ante ávidos compradores algo perdidos y desorientados ante la numerosa cantidad de ofertas.

Los servicios de telecomunicaciones a los que el ciudadano británico puede acceder desde su ordenador, por pequeño que éste sea, convierten al modem en una golosina difícilmente rechazable. Entre

estos servicios, destacan los de Microlink Prestel y Micronet 800. La utilización de estos servicios conlleva amplias posibilidades, entre las que destacan: Telex, Mailbox, Telesoftware, Base de datos, Bolletín Board...

En otro orden de cosas, el CAD/CAM fue otro de los atractivos de la edición de este trimestre del **Amstrad Computer Show**. Digitalizadores y plotters, con una calidad gráfica muy alta, constituían el otro blanco de la curiosidad del público.

En cuanto a software, nada nuevo bajo el Sol, salvo un editor de sprites y un juego conversacional de Interceptor software con una calidad gráfica muy superior a lo visto hasta la fecha.

Volviendo al hard, un nuevo periférico, el interface TWO, de Romantic Robot, que permite obtener cualquier juego, grabarlo en disco o cinta, modificarlo y un

largo etcétera. Un buen periférico que esperemos sea pronto comercializado en España.

En general, la feria constituía una plataforma para distribuidores y comerciantes, en un país donde la informática ha dejado de venderse en tiendas especializadas y ha pasado a las tiendas de electrodomésticos y a la venta por correo.

A pesar del ambiente comercial, todos se volcaron en nuestra presencia, lo que, aparte del orgullo personal que esto significa para nuestra revista, hace patente la madurez que está alcanzando el mercado informático en nuestro país, mercado, por otra parte, hasta hace muy poco bastante ignorado.

La semana que viene dedicaremos un amplio espacio a esta feria, cargada de novedades, en la que analizaremos a fondo todas las posibilidades de estos equipos.

ME DAS TU DIRECCION

Hasta ahora hemos estado hablando de que la misión de la memoria dentro de cualquier ordenador que se precie es recordarle a la CPU (Unidad Central de Proceso) del estado en que se encuentran sus «tareas».



Para tal fin anotaba en ellas los trabajos que ya ha realizado, los que todavía le quedan pendientes y, además, todos los datos que un programa, o conjunto de instrucciones, tenga que manejar o «tratar».

Así pues, habíamos quedado que dentro de la memoria de nuestro **Amstrad** van a estar guardados o «almacenados» un montón de números que representarán cosas muy distintas.

Alguno de ellos será un «código» que simboliza una instrucción Basic. El intérprete, que ya está incorporado en la memoria **ROM** de la máquina, se dará cuenta de ello y traducirá este número a un conjunto de instrucciones, en un lenguaje que ya el ordenador sí es capaz de entender, y tranquilamente las ejecuta. Tales instrucciones están compuestas totalmente por un conjunto de **ceros y unos** y es lo que conocemos como «lenguaje máquina».

Otros números, de todos los que conviven en la memoria de un **Amstrad**, representan solamente eso, un número que responde al valor que toma una variable de tipo numérico, claro, en un instante en concreto. De aquí es de donde los va cogiendo o «leyendo» la Unidad Arimético-Lógica para después operar como ellos y devolver el resultado obtenido tras la operación al espacio reservado en la memoria para contener lo que «vale» la variable correspondiente.

Por último, diremos que hay otros números que representan 1 de los 255 símbolos ya definidos que posee el juego de caracteres del **Amstrad**. Son lo que hemos llamado en otras ocasiones anteriores los códigos **Ascii** de los mismos. Así que cada uno de los signos que el ordenador utiliza para comunicarse con nosotros, cifras, letras, símbolos de puntuación y hasta algún que otro **hombrecillo** corriendo que bien podría servirnos como gráfico de cualquiera de nuestros queridos y adorados juegos de arcade, está asociado a un número que es su código **Ascii**.

Como hemos visto con todo lo anterior, una cosa debe quedarnos muy clara: todo dentro del **Amstrad** está representado por números. ¿De acuerdo? Ahora bien, ya que damos por sentado que toda la memoria está llena de números, o hablando más correctamente, ya que tenemos un montón de números almacenados en alguna «dirección» o posición de la misma, puede surgirnos una pregunta: ¿Cómo se almacena un valor cualquiera en una determinada «dirección»?

Quizás, de momento, pueda parecerle esta pregunta como algo que no tiene fácil solución o, ¡quién sabe!, si para usted sea ya algo trivial y superado. En ambos casos, le rogamos que se esfuerce un poquitín y nos siga sin asustarse, si se encuentra entre los que están en el primero de los supuestos, o con interés si es del segundo grupo, ya que siempre viene bien comprobar si nuestras propias ideas son del todo válidas, o necesitan algún ligero retoque. Vamos a ello.

El interior de la memoria de un ordenador está compuesto por una multitud de «celditas» que pueden estar de dos formas diferentes: activadas o desactivadas, no pueden tomar un estado intermedio.

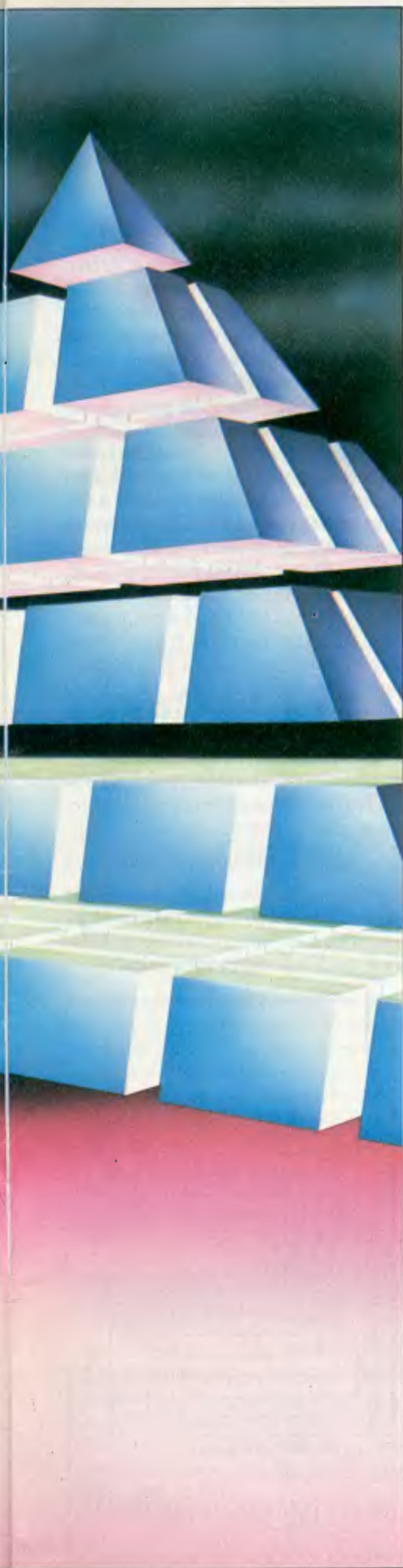
Es algo semejante a lo que ocurre con las bombillas. Cuando una recibe una corriente eléctrica adecuada, ¿qué pasa? Algo muy sencillo: la bombilla se enciende. ¿Y en caso contrario, o sea, cuando deja de circular la corriente? La respuesta es muy simple: se apaga.

Bueno, pues lo mismo ocurre con las «celdillas» de la memoria. Se activan cuando les llega una corriente y se desactivan cuando cesa o se ven afectadas por alguna de sentido contrario.

Asociemos ahora un «dígito» a cada uno de estos estados. Llamaremos 1 al de una celdilla que se encuentra activada o encendida y 0 si está apagada o desactivada. Y a base de ceros y unos vamos a construir toda nuestra teoría «informática».

Vayamos dando nombres a estos nuevos conceptos. «Bit», que es la abreviatura de Dígito Binario o Binary Digit, que dirían los ingleses, es la unidad mínima de información que un ordenador puede interpretar. Se correspon-





dería con uno de los dos estados —0, 1— que puede tomar cada una de las celdillas elementales en las que está dividida la memoria.

Bits y bytes

¿Por qué este nombre —«**Bit**»— proviene de Dígito Binario? Todos los humanos estamos acostumbrados a contar cualquier cantidad partiendo del número de veces que contiene el número diez. Por eso decimos que empleamos un sistema de numeración «**decimal**», o de base 10, ya que es en este número en que nos «**basamos**» para decir la cantidad de unidades que hay de una determinada cosa.

Y, ¿cuántos dígitos o cifras utiliza este sistema de numeración? De todos es sabido que diez, del 0 al 9. Con ellos vamos componiendo cualquier número por grande o pequeño que sea.

En cualquier sistema de numeración cada uno de los dígitos tiene dos valores muy diferentes. Uno es el propio: es el valor absoluto. Sin embargo, también tiene otro que depende de la posición que ocupe dentro del número. Por ejemplo:

32

es una cantidad que contiene 2 unidades de primer orden y 3 de segundo. En el sistema de numeración decimal es demasiado evidente que como por cada 10 unidades de primer orden se forma una de segundo, nuestro número contendrá exactamente:

$$30 + 2 = 32$$

unidades de primer orden, o unidades «a secas». En otros sistemas no lo será tanto, ya lo veremos.

Podemos extender esto. Cada unidad de tercer orden contiene diez unidades de segundo y por tanto:

$$10 \times 10 = 100$$

de primero. De modo que el número 342 podríamos representarlo como la suma de:

2 unidades de primer orden = 2.

4 unidades de segundo orden = $4 \times 10 = 40$.

3 unidades de tercer orden = $3 \times 10 \times 10 = 300$.

que nos da como resultado las 342 unidades de primer orden que todos estábamos esperando. Y así podríamos seguir indefinidamente.

Volvamos a lo nuestro. Dijimos que un ordenador sólo puede reconocer dos estados de información: 0 y 1. Pues con estos valores fabriquemos un sistema de numeración cuya base será el número 2 (en decimal) y que nos servirá para conocer el valor de cualquier número «binario» formado a partir de una combinación de ceros y unos.

Por eso, al ser el 0 y el 1 los dos dígitos que dan forma a este sistema de numeración, es por lo que hemos llamado «**Bit**» (o Dígito Bi-

Primeros PASOS

nario, como quiera) a cada uno de los dos estados —activado o desactivado— que tiene cada una de las células elementales que forman la memoria del **Amstrad**.

Ahora bien, un ordenador no suele trabajar con «bits» de información, sino que los agrupa hasta que su conjunto pueda ya significar algo que esté más al alcance de los hombres.

¿Cómo se agrupan los bits? Si fuera por nosotros no habría ningún problema, pero para nuestro ordenador, sí. El verdadero cerebro y director de todo lo que ocurre es el «**microprocesador**». El que dispone el **Amstrad** se llama Z-80 y es el que limita el número de bits que podemos agrupar. ¿Les vale con tener conjuntos de 8 bits?

Al Z-80 por supuesto que sí, perfectamente. De modo que a partir de ahora consideramos que la memoria del **Amstrad** está dividida en asociaciones de 8 celditas elementales de las que antes hablamos. Y a está combinación de 8 bits la reconoceremos como «**byte**» o en un lenguaje que entendemos un poquito mejor «**octeto**».

Bits, bytes, ¿qué tal vamos con esta nueva terminología?

Cada uno de estos bytes lo tiene el ordenador en el interior de su memoria perfectamente localizable en una posición fija: su «**dirección**».

Además la combinación de unos y ceros que forman un «**byte**», al fin y al cabo, representa un número, aunque en binario, claro. Y este número es lo que entendemos por «**contenido**» del mismo.

Así que podemos asociar dos cosas a un «**byte**». Una es su colocación o «**dirección**» dentro de la memoria y unos sirve para acceder a él cuando sea necesario. La segunda es lo que lleva en su interior: su «**contenido**» o «**dato**». Imaginamos que poco a poco todas estas ideas se irán asentando en su «memoria» y lo que aparecía casi como un acto de brujería o de magia se vaya transformando en algo más lógico y evidente y que nos deja intervenir en ello para adecuarlo a los fines que queremos conseguir.

Todo va bien, pero seguimos sin saber qué valores son los que podemos guardar en una posición de la memoria y cómo guardarlos. Vayamos por partes. Estará de acuerdo que el número más pequeño que se almacena en un byte es:

00000000

y el mayor:

11111111

¿Cuánto valen en decimal, que al fin y al cabo, es a lo que estamos acostumbrados? La transformación del primero es evidente: 0 es 0 en cualquier sistema de numeración, así que un trabajo menos.

En cuanto al otro número, analicémosle detenidamente. Como estamos empleando el sistema binario, o de base 2, podemos adaptar todo lo dicho para el decimal, pero teniendo en cuenta ahora la nueva base de numeración. Por tanto, cada dos unidades de cualquier orden forman una de orden superior, así que a trabajar, ¿vale?

Recordando, además, que cada dígito tiene un valor absoluto y otro relativo, que depende de la posición que ocupa, no debe darle muchos problemas deducir que 255 es el valor máximo que podemos guardar en un «byte». Por si acaso ojee la Figura 1 y ¡dudas fuera!

Figura uno

En ella vemos cómo se asocia a cada dígito sus dos valores, el absoluto y el relativo. Para calcular este último emplearemos siempre el mismo método que consiste en multiplicar la cifra por la base elevada a la potencia equivalente a la posición que ocupa el dígito dentro del número que estamos manejando.

¿Cómo determinamos la potencia o exponente correcto? Muy sencillamente. Tomamos como origen de cifras la que esté colocada más a la derecha de la cantidad. Como todo origen que se precie comenzará en «cero» y a partir de ahí empezaremos a contar las posiciones (o sus exponentes equivalentes).

El número binario que tenemos entre manos es el

1111111

Las unidades de primer orden son las indicadas por la primera cifra empezando por la derecha, o sea, 1. El valor decimal equivalente será:

$$1 \times 2(2^0) = 1$$

¿De acuerdo? 2 es la base y 0 el exponente que corresponde a las unidades.

Vayamos con las de segundo grado (segunda cifra comenzando por la derecha). Su valor absoluto es también 1, pero el relativo ya no es el mismo puesto que ocupa otra posición. ¿Cuál es, entonces?

$$1 \times (2^{11}) = 2$$

No se olvide de algo muy importante: ¡Estamos intentando calcular el número decimal equivalente al binario 1111111!

Podemos seguir viendo el valor relativo —en decimal, repetimos— que tiene cada dígito binario siguiendo este mismo proceso, pero variando el exponente en cada caso de acuerdo con la posición que ocupa.

Por eso se lo dejamos a usted, no se enfade. Si después sumamos todos los valores relativos obtenidos sacaremos el decimal que equivale a nuestro binario y que es, precisamente, el 255 que dijimos al principio de la historia. ¿Verdad que no supone mucha dificultad, aparte de los cálculos?

De todas formas, el Basic del **Amstrad** ya había previsto que en alguna ocasión tendríamos que realizar alguna transformación de este tipo. Y para ello nos permite utilizar un sencillo truquito que por sí solo nos ahorra trabajo. ¿Es el binario 1111111 del que queremos saber su equivalente decimal? Pues escriba:

```
PRINT &X1111111
```

y en la pantalla nos aparece el 255 que astutamente andábamos buscando.

&X es un prefijo que indica al **Amstrad** que lo que va detrás es una cantidad o constante binaria. Y él así lo interpreta y por eso hace internamente la transformación a decimal y la visualiza por medio de la instrucción PRINT. Pruebe a calcular el equivalente de algún que otro binario más y le aseguramos que puede fiarse de las respuestas que le va dando el ordenador.

Además disponemos de otra función que nos permitirá hacer todo lo contrario: transformar un número decimal a binario. O más exactamente, genera una cadena de caracteres binarios —ceros y unos— que representan el valor de un número decimal.

```
BIN$(número,cifras)
```

produce una constante alfanumérica de tantos caracteres como indique «cifras» que es equivalente al número binario obtenido a partir del decimal «número».

```
PRINT BIN$(255,8)
```

visualizará los 8 dígitos binarios que se corresponden con el número decimal 255.

```
PRINT BIN$(98,6)
```

hará lo mismo, pero esta vez con el 98, y devolviendo una cadena de seis caracteres.

Si el contenido de «cifras» no se corresponde con el número de caracteres necesarios para nuestra transformación no hay ningún problema. Si es mayor, el ordenador decide completar con ceros por la izquierda hasta llegar a las especificadas. Cuando sea menor no lo vamos a notar. El **Amstrad** hace que la expresión resultante contenga todos los dígitos necesarios. ¡Es maravilloso!

Le animamos a que haga todas las pruebas imaginables de esta función dando a los dos parámetros, «número» y «cifras», distintos valores para comprobar hasta dónde es válida la función y vea los límites que pueden alcanzar.

No pase por alto algo muy importante. La función BIN\$ produce una cadena de caracteres, luego con ella no podremos realizar ninguna operación aritmética reservada para los números, aunque el resultado de aplicar BIN\$

tenga toda la apariencia de serlo. Si no, intente hacer:

```
PRINT BIN$(3,4) × BIN$(7,4)
```

y, en lugar del 21 que esperamos, el **Amstrad** nos obsequiará con un mensaje de error.

De 0 a 255

Pero volvamos a nuestra memoria. Acabamos de comprobar que un «byte» (recuerde lo que es) puede contener números comprendidos entre 0 y 255 en decimal o, lo que es lo mismo, combinaciones de unos y ceros comprendidos entre 00000000 y 11111111. ¡Ya tenemos fijada una de las características de un byte: su «contenido».

La otra era su posición o «dirección». La memoria está organizada como una serie lineal de direcciones de un modo semejante a una larga avenida numerada. ¿Quién o cómo se fija la «cantidad» de direcciones que tiene la memoria de un ordenador?

Es bastante evidente pensar que sus características internas serán las que nos lo determinen. La Unidad Central de Proceso toma la posición donde se encuentra un dato de un «lugar» reservado compuesto por 16 bits (2 bytes) que se llama «registro de direcciones».

Y otra vez estamos en las mismas: los valores extremos que este registro puede almacenar. Pero ahora disponemos de alguna herramienta que nos facilitará el trabajo. Escriba:

```
PRINT &X0000000000000000
```

y sabremos justamente el límite inferior: 0. Era muy evidente. Luego la primera posición de la memoria que el microprocesador del **Amstrad** puede «direccionar» es la 0.

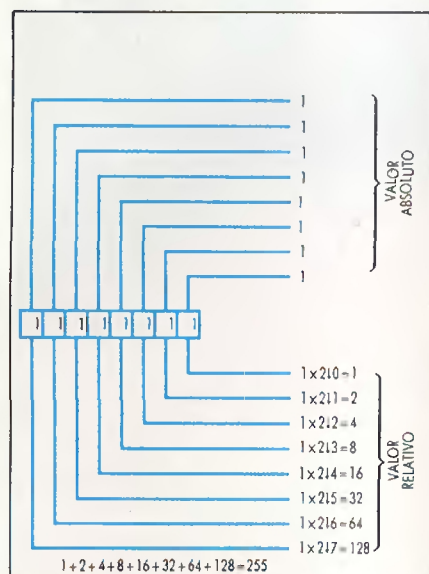


FIGURA 1: CONVERSION BINARIO-DECIMAL

Lo que ya no está tan claro es la máxima dirección a la que puede acceder, ¿verdad? Será la

1111111111111111

sí, 16 unos. ¡Qué cantidad más grande! 65535 es la posición más alta de un byte a la que CPU puede llegar por los métodos habituales de direccionamiento.

Si quiere comprobarlo, teclee:

```
PRINT BIN$(65535,16)
```

y verá que no le mentimos. Aparecen en la pantalla los 16 unos.

¿Quiere decir esto que nuestro ordenador tiene una memoria RAM de 65535 bytes? No, solamente que el **Amstrad** puede acceder directamente a datos que se encuentren en direcciones comprendidas entre la 0 y la 65535. Nada más.

¿Cuál es, entonces, la capacidad real de la memoria? Tomaremos como unidad de medida el Kbyte: las Ks que todos hemos oído alguna vez. Pero, ¿cuántos bytes tiene un K? El cálculo es elemental:

$1K = 2^{10}$ bytes

Así que la capacidad de cualquier ordenador será el resultado de dividir el número de bytes que posee entre 1024 (2^{10}) y no dará los conocidos Ks de los que podemos disponer.

O sea:

$1K = 1024$ bytes

Multiplique esta cantidad por el número de K que tiene su ordenador y obtendrá la gran cantidad de bytes, o conjuntos de ocho bits que llenan su memoria central.

Teclee ahora:

```
PRINT 65535 + 1/1024
```

y, ¿qué le aparece en la pantalla? Pues justamente los 64 K a los que puede acceder directamente el microprocesador del que está equipado el **Amstrad**. Eso no quiere decir que sólo tenga 64 K de capacidad sino que para llegar a direcciones superiores a la 65535, utilizará otros métodos que de momento dejaremos a un lado. Piense que hay modelos de 64, 128, 256 y 512 K de memoria RAM.

Acerca del POKE

Bueno, ya sabemos mucho acerca de la memoria del **Amstrad**. Sus bytes sólo pueden contener valores comprendidos entre 0 y 255 y ocupar una dirección que oscila entre 0 y 65535. Lo que no sabemos todavía es cómo introducir un valor en una determinada posición, o, ¿no es así?

POKE dirección,valor

es una orden que escribe el valor almacena-

```
10 REM PROGRAMA 1
20 ON ERROR GOTO 100
30 WINDOW #1,1,40,4,22
40 WINDOW #2,1,40,24,24
50 WINDOW #3,1,40,25,25
60 CLS
70 PRINT"DIRECCION", "CONTENIDO"
80 PRINT"-----", "-----"
90 WHILE -1
100 INPUT #2,"DIRECCION (99999-FIN)
: ",direccion
110 IF direccion<0 OR direccion>65535
THEN canal=2:ERROR 33
120 INPUT #3,"CONTENIDO: ",contenido
130 IF contenido<0 OR contenido>255 T
HEN canal=3:ERROR 34
140 POKE direccion,contenido
150 PRINT #1,direccion,contenido
160 CLS #2:CLS #3
170 WEND
180 REM Rutina Tratamiento de Error
190 IF direccion=99999 THEN END
200 CLS #canal
210 IF ERR=33 THEN PRINT #2,"DIRECCIO
N INCORRECTA":GOSUB 230:RESUME 100
220 IF ERR=34 THEN PRINT #3,"DATO INC
ORRECTO":GOSUB 230:RESUME 120
230 REM Rutina de Retardo
240 FOR i=0 TO 1000:NEXT
250 CLS #canal
260 RETURN
```

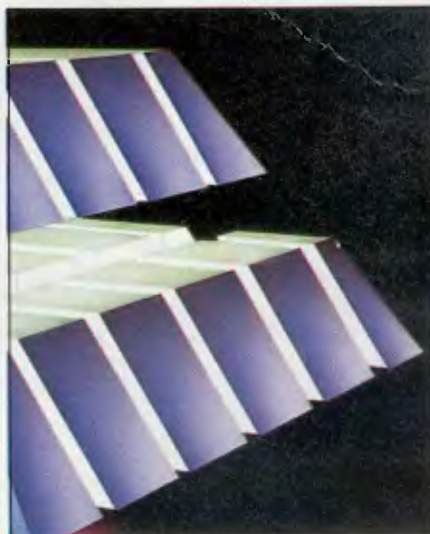
do en «valor» en la dirección indicada por «**dirección**».

«POKE» significa meter, introducir y permite escribir en una celdilla de memoria (o byte) un valor comprendido entre 0 y 255. Anteriormente ya indicamos que este valor puede representar un dato, una instrucción o un carácter Ascii.

Por ejemplo:

POKE 42321,69

está colocando en el byte situado en la posi-



Primeros PASOS

ción 42321 la letra E, que es el carácter que corresponde al código Ascii 69.

Por medio de esta instrucción podemos conseguir que el ordenador realice las cosas más inverosímiles: iluminar la pantalla, generar, ruido, etc... Sólo necesitamos conocer la dirección y el valor exacto que queremos introducir. De momento nos conformaremos con ir «**metiendo**» en direcciones cualesquiera los valores que se nos ocurran: puede suceder hasta que el ordenador se canse y se pare.

Pero no hay cuidado, lo desenchufamos y punto. ¡A empezar de nuevo!

Una pregunta más. ¿Cómo ha de ser la memoria sobre la que vamos a modificar el contenido de un byte? La memoria ROM es de sólo lectura así que de ella sólo podemos sacar información y, por tanto, no podemos almacenar nada allí. La instrucción POKE no la causa ningún efecto, luego queda descartada.

La RAM es una memoria de lectura y escritura, pues ya lo tenemos. En sus bytes no sólo podremos husmear para ver qué se guarda allí, sino que además tenemos la posibilidad de cambiar el valor de los datos que contienen. POKE es la llave maestra para hacerlo.

Recuerde: ¡Sólo podemos emplear la sentencia POKE para modificar direcciones de la memoria **RAM**. Con la **ROM** no podemos hacer nada!

Bien, le dejamos con un sencillo programa que le permitirá variar lo que hay en cualquier posición comprendida entre 0 y 65535. El ordenador no nos permitirá poner como parámetros de POKE valores que se salgan de los márgenes establecidos. Si teclea:

POKE 70000,3

le está diciendo al **Amstrad** que almacene un tres en la dirección 70000. ¿Es posible hacerlo?

¡Evidentemente, no! El **Amstrad** sólo puede acceder a posiciones de memoria que no superen la 65535. ¿No es así?

Del mismo modo cuando escriba:

PKE 20000,1000

está queriendo almacenar en el byte 20000 el número 1000. Pero en 8 bits sólo podemos guardar números comprendidos entre 0 y 255 así que la tarea que le estamos encomendando al **Amstrad** no se puede realizar.

En ambos casos nuestro ordenador nos lo advertirá sacando un «**aviso**» de que algo anda mal. Así que ¡jojo!

Imaginamos que no encontrará ningún problema en comprender cómo funciona. Pues entonces, ¡manos a la obra y duro con él!

Por José María Díaz

Si nuestros lectores observan algún que otro juego comercial, se percatarán enseguida de que, la mayoría de ellos, incorporan una subrutina en la que el ávido jugador elige las teclas que va a utilizar para moverse a través de él. Análisis propone ésta.



Para aquellos desafortunados que no posean un joystick, el teclado posee una capital importancia para poder jugar a algo. Además, si uno es programador, y le gusta hacérselos «motu proprio», no estaría de más tener disponible la famosa rutina de «propósito general», que le sirva para todos.

Este pequeño programa de análisis responde a este enfoque. Partiendo de la suposición de que se van a usar 4 teclas de movimiento, el programa pregunta cuáles son y luego las

utiliza para mover por la pantalla una «A» en representación del gráfico real que se usará en un juego.

Vamos a estudiar brevemente la estructura del programa.

Líneas 10-50: definición de variables generales y modo de pantalla.

Líneas 60-120: llamadas a las distintas subrutinas que configuran el programa.

Líneas 140-170: lectura de las cuatro direcciones de movimiento, e introducción de las mismas en la matriz «direc\$».

Líneas 180-220: impresión en pantalla de los nombres de las cuatro direcciones.

Líneas 230-300: lectura de las teclas escogidas (260) y asignación de cada una a un elemento de la matriz «teclas\$» (270).

Líneas 310-350: se comprueba que el

usuario está satisfecho de las teclas elegidas y de que no ha habido error. Si éste existiera, en el bucle principal se comprueba el valor de la variable «resp\$» (línea 100) y se actúa en consecuencia.

Líneas 360-450: una vez escogidas las teclas, esta subrutina realiza una animación muy primitiva para que comprobemos que funciona. La única línea compleja es la 420; el animamiento de las sentencias IF...THEN...ELSE sirve para que sólo se realicen comprobaciones hasta que se encuentre una igualdad entre la tecla pulsada y las almacenadas en la matriz «tecla\$». Se sale del bucle cuando se pulse la tecla «f», de «fin».

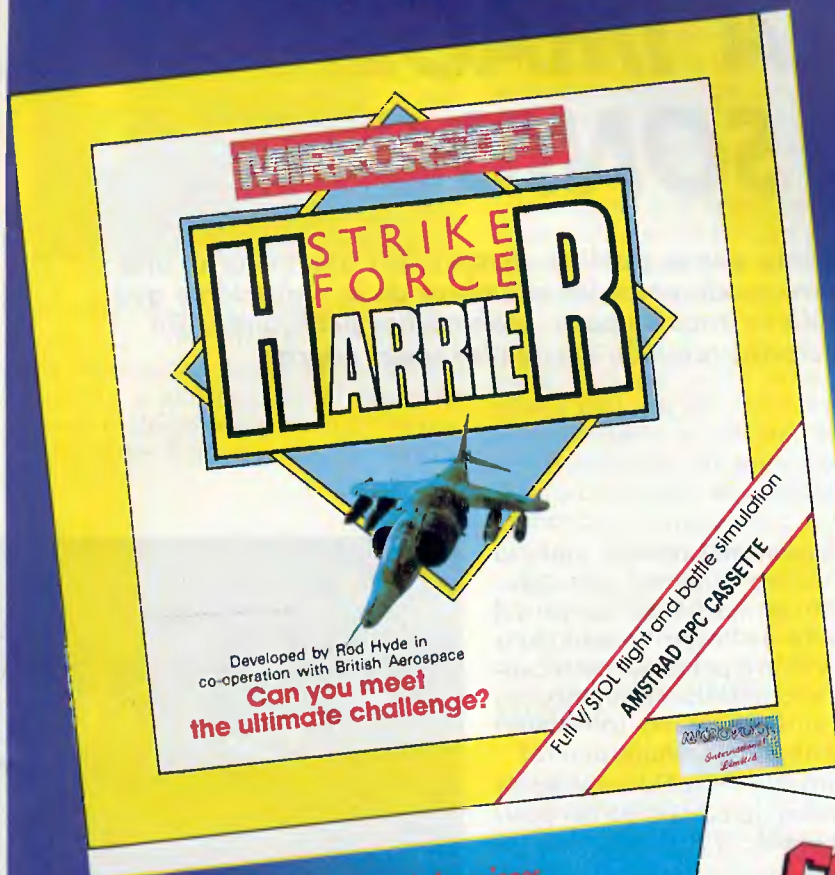
Esta rutina, eliminadas las sentencias REM y con nombres de variables más cortos, ocupará poca memoria y será de gran utilidad.

```
10 REM PANTALLA Y VARIABLES
20 MODE 1
30 ndatos%=4:DIM tecla$(ndatos
%),direc$(ndatos%):resp$="":te
clafin$="f"
40 hpos%=10:vposi%=10
50 GOSUB 140:REM LECTURA
60 CLS
70 GOSUB 180:REM IMPRESION
80 GOSUB 230:REM LECTURA TECLA
90 GOSUB 310:REM RESPUESTA COR
RECTA?
100 IF resp$="N" THEN 60
110 PRINT"TECLADO DEFINIDO"
```

```
120 GOSUB 360:REM MOVIMIENTO
130 END
140 REM LECTURA
150 FOR i%=1 TO ndatos%:READ d
irec$(i%):NEXT
160 RETURN
170 DATA "derecha?","izquierda
?","arriba?","abajo?"
180 REM IMPRESION
190 FOR i%=1 TO ndatos%
200 LOCATE 15,10+i%:PRINT dire
c$(i%)
210 NEXT
220 RETURN
230 REM LECTURA TECLAS
240 FOR i%=1 TO ndatos%
250 LOCATE 26,10+i%:PRINT "_"
260 resp$=INKEY$:IF resp$="" T
HEN 260
270 tecla$(i%)=UPPER$(resp$)
280 LOCATE 26,10+i%:PRINT resp
$
290 NEXT
```

```
300 RETURN
310 REM RESPUESTA CORRECTA?
320 LOCATE 15,20:INPUT "Correc
to (S/N)";resp$
330 resp$=UPPER$(resp$)
340 IF resp$()="S" AND resp$()=
"N" THEN 310
350 RETURN
360 REM MOVIMIENTO
370 CLS:resp$=""
380 WHILE resp$()teclafin$
390 LOCATE hpos%,vposi%:PRINT
A"
400 resp$=INKEY$:IF resp$="" T
HEN 400
410 resp$=UPPER$(resp$)
420 IF resp$=tecla$(1) THEN hp
os%=hpos%+1 ELSE IF resp$=tecl
a$(2) THEN hpos%=hpos%-1 ELSE
IF resp$=tecla$(3) THEN vposi%
=vposi%-1 ELSE vposi%=vposi%+1
430 LOCATE hpos%,vposi%:PRINT
A"
440 WEND
450 RETURN
```


¡¡ATREVETE!!



DEMUESTRA TU HABILIDAD
ENFRENTÁNDOTE A LOS
DOS «SUPER-COLOSOS» E
INTENTA LLEGAR AL FINAL
¿Crees que lo conseguirás?

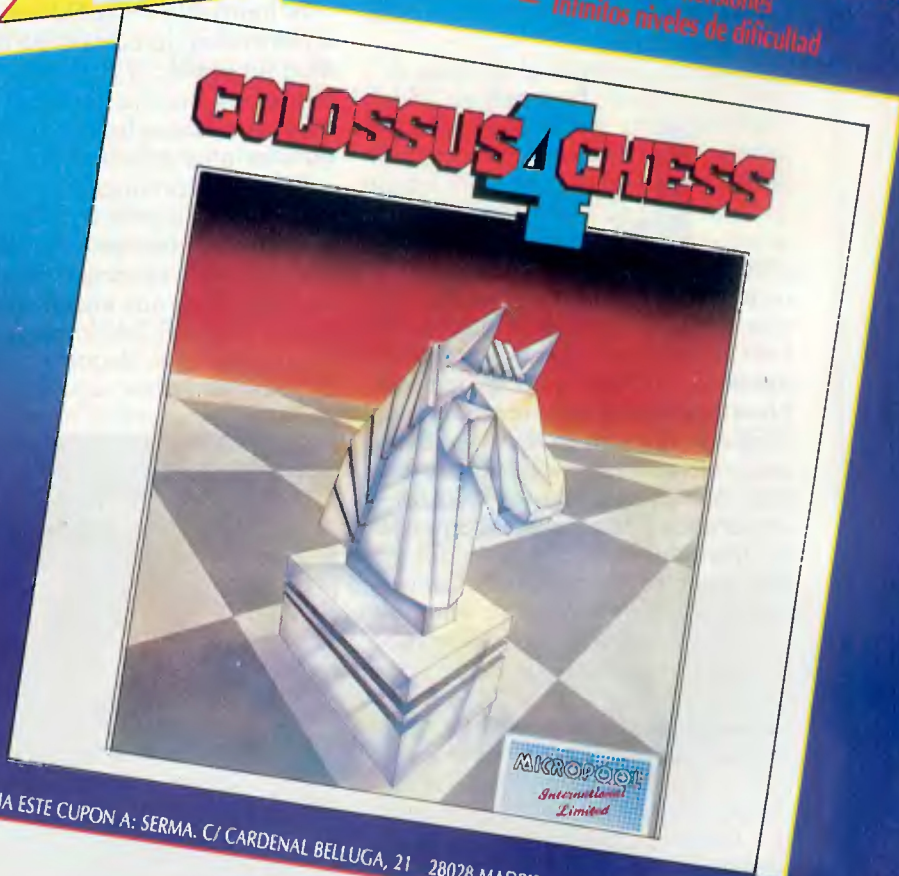
- Para llegar al nivel de los campeones
- Dos y tres dimensiones
- Infinitos niveles de dificultad

Demuestra tu habilidad derribando los aviones
enemigos para poder destruir su base.

COLOSSUS
AMSTRAD: DISCO: 3.600 ptas.
CASSETTE: 2.500 ptas.

HARRIER
AMSTRAD: DISCO: 3.600 ptas.
CASSETTE: 1.900 ptas.

TODOS LOS PRODUCTOS EDITADOS
POR EL GRUPO MICROPOOL
SON UNA EXCLUSIVA DE SERMA



RECORTA Y ENVIA ESTE CUPON A: SERMA. C/ CARDENAL BELLUGA, 21 28028 MADRID. Tels. 256 21 01/02 - 256 50 06/05/04

COLOSSUS ☐ ☐

HARRIER ☐ ☐

NOMBRE Y APELLIDOS: _____

DIRECCION: _____ CODIGO POSTAL: _____

POBLACION: _____ PROVINCIA: _____

FORMA DE PAGO: TALON BANCARIO ☐ CONTRA REEMBOLSO ☐

SONIMAG 86 TODA LA IMAGEN Y EL SONIDO

Detrás de cada artículo, de cada texto que se publica, siempre hay una historia, una anécdota que por lo general siempre queda entre los miembros de la redacción y que algún día, aunque sólo sea por divertirnos un poco, acabaremos publicando. Sin embargo, entre estas historias también existen leyendas negras.

La edición de este año de Sonimag, al menos a priori, resultaba más que interesante. Así que, movidos por el noble fin de informarnos para informar, cogimos nuestros bártulos y a Barcelona. Nuestro primer error: contamos con IBERIA, y me explico. Una vez realizado el reportaje, después de dos días de trasiego, más de 300 fotos disparadas y tres pares de alpargatas gastadas contra el suelo del pabellón cuatro del Palacio de Congresos, embarcamos nuestro equipaje en uno de los pájaros de hierro de la insigne empresa que surca nuestros cielos. A la llegada a Madrid la sorpresa: ¡el equipaje había desaparecido! Nuestra persecución continúa durante dos semanas, a través de todo el personal de IBERIA resultó infructuosa; explicamos la importancia del contenido, hablamos con distintos aeropuertos de España, pero nada. La respuesta era siempre la misma.

Nuestras últimas noticias sobre la maleta viajera confirmaban su presencia en el muelle de carga del aeropuerto de Palma. Reenviada a Madrid volvió a perderse misteriosamente y ahora puede encontrarse en Zurich, en Pekín, o en Tahilandia, ¡lástima no haber estado dentro!

Al margen de ineptitudes, ineptos o ineficacias, la edición 86 del Sonimag ha venido a confirmar que las cosas, en España pueden hacerse bien; sólo hacen falta dos cosas: organización y esfuerzo.

El primer contacto con el Sonimag fue decepcionante. Tras de subir los escalones de tres en tres hasta el nivel 10, donde se encontraba el stand del **AMSTRAD** nos encontramos con que el nuevo PC de la compañía de Sagar no había llegado a España. Después de haber superado el con-



siderable trago que nos supuso esta ausencia, comenzamos nuestro paseo por toda la feria en busca de más novedades.

La estrella de la feria, no olvidemos que SONIMAG se concibió para la imagen y el sonido, fue el Compact-Disc. Numerosos fabricantes de HIFI se han lanzado sobre este nuevo reto en busca del «elixir del sonido».

Llevado a nuestro campo, el de los ordenadores, esta revolución en la grabación del sonido aporta, como ya lo hiciera el cassette, un potente impulso en el almacenamiento de información.

Entre estos equipos es de destacar



el adoptado como estandar bajo técnica CD-1, que ofrece la posibilidad de albergar en su interior imágenes, sonidos e información. El siguiente cuadro ofrece un resumen de las posibilidades tanto en imagen, texto y sonido.



FORMATO FISICO

Memoria: 650 megabytes.
Velocidad de lectura: 75 bloc/sec.
Bloque: 2.352 bytes.
Subíndice: 8 bytes.

INFORMACION ALMACENABLE

Información en audio y datos de ordenador.

AUDIO

	canal
CD-Digital audio Estereo	1
Música alta fidelidad	2 y 4
Música fidelidad media	4 y 8
Diálogo	8 y 16
Duración por canal: 70 minutos.	

VIDEO

Resolución de referencia. 768x560

El Compact Disc Interactivo es aun una técnica en sus inicios. Sin embargo, y dada la vertiginosa evolución de la tecnología, no sería de extrañar que dentro de poco, de muy poco, la televisión, la cadena de música y el ordenador queden fundidos en un mismo macroequipo, que todo lo sabe y todo lo dice. Es lo que verdaderamente podríamos considerar como FUTURO. Las primeras aportaciones de esta tecnología, en la que la cabeza de lectura se transforma en un haz láser, han comenzado a ofrecer sus frutos y prometen entrar con fuerza en el mercado.

Por el momento ya se están comercializando los primeros CD-ROM (*Compact Disc Read Only*) con una capacidad que oscila entre los 500 y los 1.000 megas, ¡asombroso! Sólo un pequeño problema, al menos por el momento, de estos discos: sólo se puede leer, el escribir resulta demasiado costoso para el usuario medio y el borrar es imposible.

Siguiendo en el campo de la HIFI, tuvimos ocasión de observar la nueva línea de productos de bajo precio. En este terreno son de destacar dos compañías, INVESTRONICA y **AMSTRAD**, si leyeron bien.

La entrada de Sugar en este nuevo campo se realiza como ya hiciera en el terreno de los ordenadores, precios de risa para productos de gran calidad.

La gama de equipos presentada por Amstrad, va desde una minicadena con amplificador, sintonizador, plato y ¡doble pletinal!, a un precio de 29.000 pesetas hasta otro con compact disc por algo más de 85.000 pesetas.

La otra gran estrella de la feria fue la televisión vía satélite, que según todos los indicios va a constituir el plato fuerte de la alta tecnología durante las próximas navidades.

Nuevos equipos con un costo bastante más bajo al de las primeras antenas, prometen llevar la televisión europea hasta los hogares más modestos, téngase en cuenta que con una antena son muchos los vecinos que pueden disfrutar de la

TV vía satélite. Entre todos estos equipos es de destacar el distribuido por Microbyte, con un costo total de 375.000 pesetas, y que ofrece la posibilidad de seleccionar cualquiera de los 12 canales que emite el TELESAT.

Sonimag 86

24 Salón Internacional de la Imagen, el Sonido y la Electrónica

Y de informática, ¿qué?

A parte de la comentada ausencia del PC de **Amstrad**, tuvimos la posibilidad de observar una marcada tendencia «PC», en los nuevos modelos del mercado. Al otro lado de los increíbles «AMIGA» de Commodore y 520 ST de ATARI, cuya presencia no acaba de cuajar en el mercado, el sector parece agarrarse al compatible estándar de IBM, eso sí, cada vez con mayores prestaciones, mejores precios, y en el caso de **Amstrad**, incorporando un sistema operativo, el GEM, que quita el hipo.

En cuestión de software, al menos para **Amstrad**, estaban todos. Drosoft con sus programas baratos y ca





da vez con más calidad. Erbe con un stand hiperfuturista y una enormidad de monitores donde disfrutar con sus juegos. Mindgames con tres nuevos juegos importados para los CPC, Glass, Souls of darkon y Core. Así como Software Center que aparte de su gama de productos para **Amstrad** presentaba software de fabricación propia para el AMIGA de Commodore.

A nivel de equipos domésticos, la presentación del nuevo Spectrum +2, primer equipo desarrollado por Sinclair, tras la compra de la marca por parte de **Amstrad**, constituía el principal atractivo de los **microjugones**. La nueva máquina que como innovación fundamental, respecto a sus antecesores, presenta la incorporación de un teclado absolutamente profesional, un cassette, y salida de



joystick, tiene como mayor virtud el poder aprovechar casi todo el excelente software diseñado para el popular Spectrum 48.

Nuevas iniciativas

Especialmente interesante resulta la iniciativa presentada por Idealogic, para el desarrollo del lenguaje Logo. Durante el presente año han trabajado en la creación de aulas informáticas dentro del ambicioso proyecto bautizado como Centre Logo que culminará con la realización de una maqueta urbana, donde todos sus elementos estarán controlados por ordenador. A parte de otras actividades, destaca la publicación de una revista-folleto dedicada integra-

trad. En la próxima semana estudiaremos este fabuloso equipo a fondo y analizaremos todas sus posibilidades.

La ya conocida compañía, por su distribución de Sinclair, INVESTRONICA, ha sido una de las últimas en subirse al carro de la compatibilidad PC. Durante el transcurso de la Feria, fue presentada la nueva serie PC-INVES, que con sus distintos modelos pretendía satisfacer a todos los usuarios. Básicamente la serie se apoya en dos equipos, el PC-256X y el PC-640A, de los que emanan distintas configuraciones.

El PC 256 X está basado en el 8088 de INTEL, lo que, aunque garantiza casi totalmente una compatibilidad prácticamente absoluta con su gemelo de IBM, puede considerarse ya como un procesador algo anticuado por su velocidad. Particularmente prefiero el 80086, aun contando con los pequeños problemas que puedan surgir con algunos programas que corran bajo el estándar.

El modelo PC-640A por el contrario, con un intel 80286, sí que puede considerarse como un auténtico monstruo de la informática personal. Naturalmente se garantiza la com-



mente al Logo y que puede conseguirse escribiendo a Idealogic (Etenza 218. Barcelona).

Entre los productos de hardware, resultaba especialmente atractivo el convertidor de monitor a TV íntegramente diseñado y fabricado en España por MHT. Mediante este virtuoso aparatito podremos ver cualquier programa de televisión o película de vídeo en el monitor de nuestro **Ams-**

patibilidad con el IBM PC AT, un equipo que revolucionó la informática en su día.

El standar MSX

La gama MSX, la esperanza japonesa que intentó hacerse con el mer-



cado y que se ha visto doblegada ante el avance de **AMSTRAD**, **SINCLAIR** y **COMMODORE**, presentaba no obstante un software de alta calidad así como unos periféricos de una potencia más que considerable.

La especial atención que en este stander se está contemplando por el tratamiento de la imagen, ha alcanzado su punto máximo con el digitalizador de vídeo, con resultados espectaculares en trabajos de creación de vídeos, pantallas de presentación, etc...

De entre el software presentado, dos atraían fundamentalmente la atención del público. El primero, distribuido por Idealogic, era un curso completo de aerobio, en el que una graciosa muñeca va realizando una serie de ejercicios con una música de fondo que podíamos calificar de sinfónica. El otro programa, un paquete musical integrado, sólo merece la calificación de maravilloso. La diferencia entre un sintetizador y un MSX con un midi y este programa, puede considerarse nula.

Software de gestión

Dentro de software de gestión, lo mejor es prácticamente callar, no, no es que fuese malo, sino todo lo contrario: era tal el volumen de programas, y la profundidad de sus posibilidades, que harían falta muchas revistas para exponer todos con detalle.

Era curioso observar cómo este software ha sido absorbido prácticamente por **AMSTRAD**. Numerosas casas nacidas alrededor de los CPC y del PCW, han desarrollado unas aplicaciones que están resolviendo la práctica totalidad de los problemas

que surgen en torno a la pequeña y mediana gestión, y, lo que es más importante, con una producción íntegramente española. ¡Ya era hora!

La evolución de este software ha ido desde las aplicaciones de carácter muy general, procesadores de texto, bases de datos y hojas de cálculo, hasta la última línea de actuación, provocada por la saturación del mercado, cuya característica general es la particularización de las aplicaciones hacia problemas muy concretos de la gestión. Programas para la gestión de vídeo clubs, gestores de efectos comerciales... En resumen, un software prácticamente a medida, pero a un precio muy pró-

ximo al de las aplicaciones más generales.

Paralelamente a estos programas están surgiendo paquetes que permiten al usuario, construir sus propias aplicaciones. Estos programas, bastante difíciles de utilizar, vienen a demostrar que el nivel informático está aumentando; naturalmente, no me refiero al del profesional de la informática, sino al ciudadano medio que busca en los ordenadores una solución a sus problemas cotidianos.



En resumen: Sonimag no ha perdido ni un ápice de interés para el aficionado en cualquiera de los campos que cubre, y lo que es más importante, continúa estando al día. Lástima que tanta tecnología no se aproveche ni para encontrar una maleta.

CAMINO CRITICO

Speed Soft

Son muchas las tareas que realizamos diariamente, de una manera rutinaria, y sobre las que nunca nos hemos parado a pensar si hay una forma más razonable o productiva de hacerlo. Asimismo alguna vez surge algún proyecto fuera de lo común y después de pasar bastante tiempo con él, desgraciadamente, no siempre resulta como nosotros esperamos. Todos sabemos que una buena organización es fundamental para conseguir el éxito en cualquier empresa que llevemos a cabo. Con mala organización difícilmente podremos realizar, repartir y coordinar las distintas tareas de las que se compone el proyecto.



Esta dificultad crece a medida que aumenta el número de tareas.

En fin, no es nuestra intención desanimaros ante los distintos proyectos (*grandes o pequeños*) que tengáis que realizar a lo largo de vuestra vida, sino todo lo contrario; ayudaros.

Por eso vamos a daros el nombre a la palabra clave que hará, dentro de lo posible, que vuestros proyectos se realicen de la forma más racional: PLANIFICAR.

El primer paso para estudiar un proyecto es hacer una red o grafo del mismo.

Una red es una representación gráfica de la actividad o proyecto a realizar.

En el grafo, cada tarea se representa por un círculo, aunque también hay otros autores que lo representan simplemente por un punto e incluso por un cuadro. Nosotros utilizaremos el círculo. A estos círculos se los denomina nudos de la red.

Los nudos están unidos entre sí por líneas rectas. Si de un nudo sale más de una recta hacia otros nudos significa que la tarea que ese nudo representa ha de terminar para que las de los otros nudos empiecen.

De igual manera si a un nudo llegan varias líneas, hasta que no terminen todas las tareas que esos nudos representan, no podremos comenzar esta última.

Nótese que hemos mencionado «**llegan**» y «**salen**» pero, ¿QUE LINEAS ENTRAN Y QUE LINEAS SALÉN?

Es muy sencillo. Tomaremos el convenio de que el nudo más a la izquierda es el comien-

zo e iremos de izquierda a derecha hasta llegar al último nudo, que será el final del proyecto.

Si un nudo es condición para que otro empiece se llama antecedente.

Todos los nudos tienen antecedentes menos el comienzo.

Si un nudo es antecedente de otro este último es consecuente del primero.

Por supuesto, que otro dato indispensable es la duración de las tareas.

Cada tarea tiene una duración determinada que nosotros hemos de saber de antemano (*o estimarla lo más correctamente que podamos*) para introducirla en el programa cuando éste nos lo pida.

Esta parte es fundamental porque sobre ella se desarrolla el CAMINO CRITICO.

¿Qué es el camino crítico?

Imaginemos que tenemos diez tareas diferentes con un tiempo máximo de realización para cada una, como es lógico unas dependerán de otras para poder iniciarse.

Quien todavía no haya comprendido la filosofía de COMPLANIFICAR pensará que esto es una tontería. Bastará con iniciar todas las tareas lo antes posible y así se terminará lo antes posible.

Bien, esto obviamente es cierto, pero...

Supongamos que somos ingenieros de una compañía y nos mandan construir un puente en el mínimo tiempo posible y, esto es importante, con el mínimo coste. Supongamos también que en determinada fase de la construcción tenemos que traer unas estructuras desde una ciudad distante de la que nos encontramos y que esas estructuras deben permanecer en su transporte especial hasta el mo-



GESTION

mento de su colocación, por un equipo también especializado.

Si iniciamos todas las tareas lo antes posible probablemente nos sucederá que las estructuras, el camión que las transporta y el equipo especializado en colocarlas estén esperando varios días hasta que finalicen las tareas precedentes, ya que a ellos también los hemos llamado lo antes posible. Por supuesto, el transportista y el equipo de colocación cobran todos estos días aunque no trabajen, con lo que no se puede decir que hacer esto sea muy rentable para nuestro proyecto.

Se trata, pues, de conocer las tareas o nudos donde debamos poner el máximo esfuerzo en acabar a tiempo para que el proyecto termine en un plazo fijo. En las demás tareas nos interesará saber el margen de varianza o tardanza que tenemos sin que ello afecte a la finalización del proyecto en el plazo previsto. Este margen de tardanza se denomina **HOLGURA** en la terminología PERT (*Técnica de Revisión y Evaluación de Proyectos*).

El programa calcula tres tipos de holgura; la **HOLGURA INTERFERENTE**, la **HOLGURA INDEPENDIENTE**, y la **HOLGURA TOTAL**.

Las dos primeras son el resultado de recorrer la red desde el primer nudo hasta el último y al contrario, y hallar unas holguras específicas de cada actividad. No queremos profundizar mucho en los mecanismos en los cuales se basan los cálculos de las holguras ni en el significado tangible de las dos primeras porque nos llevaría bastante tiempo y desde luego se necesitarían bastantes más páginas de las que disponemos.

Por otra parte, lo que en realidad le interesa conocer al usuario del programa es la holgura total y el camino crítico. Quien desee profundizar más en el tema tiene a su disposición una variada gama de libros sobre el mismo en cualquier librería técnica. La holgura total nos da el retraso que puede tener una tarea sin afectar a las demás en la conclusión del proyecto.

Si calculamos las fechas primera y última (*osea la fecha de empuje más pronta y la fecha de empuje más tardía posible*) de todos los nudos de un proyecto se hallará al menos un camino en el cual las duraciones de ambas sean iguales para cada uno de los nudos que ese camino recorre en nuestra red. Ese es precisamente el llamado camino crítico.

La duración de todas las tareas del camino crítico es igual a la duración total del proyecto y cualquier retraso de una de ellas provocará un retraso en el proyecto.

En cuanto al manejo del programa es muy simple y en todo momento podéis acceder al menú principal que os muestra todas las opciones disponibles en el mismo.




```

10 REM *****
****
20 REM
30 REM
40 REM
50 REM CAMINO CRITICO
60 REM
70 REM POR
80 REM
90 REM SPEED SOFT
100 REM
110 REM PARA
120 REM
130 REM
140 REM MICROHOBBY AMSTRAD
150 REM
160 REM
170 REM *****
****
180 MODE 2:LOCATE 1,12:PRINT"SI QUI
ERES VER LAS INSTRUCCIONES PULSA -I-
,CUALQUIER OTRA TECLA PARA COMENZAR
"
190 A$=UPPER$(INKEY$)
200 IF A$="" THEN 190
210 IF A$="I" THEN GOSUB 5420
220 MODE 1:PEN 1
230 WINDOW#2,1,40,23,25: BORDER 4:CLS
240 DIM matriz(20,31),nomb$(20),com
$(200),come$(20):n1=0:DEG
250 mp=6:mm=4:mr=4:md=5
260 DIM mu$(MP),mp$(mp),mm$(MM),md$(
MD),mr$(MR)
270 GOSUB 4980
280 CLS:CLS #2:PRINT"basic 1.1":NEW
290 :
300 REM salir
310 :
320 sal=0:RETURN
330 REM salir del programa
340 :
350 sal=0:RETURN
360 :
370 REM CALCULAR
380 :
390 matriz(1,23)=0:matriz(1,24)=mat
riz(1,31)
400 n2=1:n3=0
410 WHILE n2<n1
420 FOR j=2 TO n1
430 n3=0:IF matriz(j,24)<>0 THEN GO
TO 580
440 FOR j=1 TO n1
450 IF matriz(j,j)=1 THEN n3=n3+1
460 NEXT
470 n4=0:IF n3>n2 THEN GOTO 580
480 FOR j=1 TO n1
490 IF matriz(j,j)=1 AND matriz(j,2
4)<>0 THEN n4=n4+1
500 NEXT
510 IF n4<n3 THEN GOTO 580
520 dis=0
530 FOR j=1 TO n1
540 IF matriz(j,j)=1 THEN dis1=matr
iz(j,24)
550 IF dis<dis1 THEN dis=dis1
560 NEXT
570 matriz(1,23)=dis:matriz(1,24)=d
is+matriz(1,31):n2=n2+1
580 NEXT
590 WEND
600 N4=0
610 FOR I=1 TO N1
620 N3=0
630 FOR J=1 TO N1
640 IF MATRIZ(J,I)=0 THEN N3=N3+1
650 NEXT
660 IF N3=N1 THEN N4=I
670 NEXT
680 N2=1:N3=0:MATRIZ(N4,25)=MATRIZ(
N4,23):MATRIZ(N4,26)=MATRIZ(N4,24)
690 WHILE N2<N1
700 FOR I=1 TO N1
710 N3=0:IF MATRIZ(I,26)<>0 THEN GO
TO 870
720 FOR J=1 TO N1
730 IF MATRIZ(J,I)=1 THEN N3=N3+1
740 NEXT
750 IF N3>N2 THEN GOTO 870
760 N4=0

```

TABLA DE SUBROUTINAS

10-170	Inicialización.
180-230	Salidas.
250-900	Cálculos del programa.
910-970	Datos básicos del programa.
980-1140	Dibuja nudos.
1150-1340	Da la posición del nudo a imprimir.
1350-1500	Introducir antecedente.
1510-1710	Dibuja antecedente.
1720-1790	Controla si se van a introducir más nudos.
1800-1950	Comentario.
1960-2120	Grabar datos.
2130-2350	Cargar datos.
2360-2480	Nuevo proyecto.
2490-2610	Consigue consecuentes.
2620-2700	Selecciona nudo.
2710-2780	Cambiar duración.
2790-2870	Modificar comentario.
2880-3040	Modificar antecedentes.
3050-3110	Cambia nombre de un nudo.
3130-3250	Menú modificar.
3270-3380	Imprime menús.
3390-3490	Insertar.
3510-3620	Menú modificar datos.
3630-3780	Borra nudo.
3790-3930	Menú de resultados.
3940-4080	Camino crítico.
4080-4500	Visualiza resultados.
4250-4560	Pausa.
4580-4780	Diagrama de barras.
4860-5090	Menú principal.
5100-5220	Mueve opciones de los menús.

```

770 FOR J=1 TO N1
780 IF MATRIZ(J,I)=1 AND MATRIZ(J,2
5)<>0 THEN N4=N4+1
790 NEXT
800 IF N3>N4 THEN GOTO 870
810 DIS=0:dis1=0
820 FOR J=1 TO N1
830 IF MATRIZ(J,I)=1 THEN DIS1=MATR
IZ(J,25):fg=fg+1
840 IF fg=1 THEN dis=dis1 ELSE IF d
is<=0 OR dis1<DIS THEN dis=dis1
850 NEXT
860 MATRIZ(I,26)=DIS:MATRIZ(I,25)=A
BS(DIS-MATRIZ(I,31)):N2=N2+1
870 NEXT
880 WEND
890 FOR I=1 TO N1
900 MATRIZ(I,27)=ABS(MATRIZ(I,23)-M
ATRIZ(I,25))
910 NEXT
920 FOR I=2 TO N1
930 N2=0
940 FOR J=1 TO N1
950 IF MATRIZ(j,j)=1 THEN N3=MATRIZ
(j,23)
960 IF j=1 THEN n2=n3
970 IF N2>N3 THEN N2=N3
980 NEXT
990 IF n2=0 THEN MATRIZ(I,28)=0

```

```

1000 MATRIZ(I,29)=ABS(MATRIZ(I,27)-
MATRIZ(I,28))
1010 NEXT
1020 RETURN
1030 :
1040 REM datos basicos
1050 :
1060 CLS#2:WHILE INKEY$<>"":WEND
1070 INPUT#2,"tiempo de realizacion
de la tarea";matriz(n1,31)
1080 INPUT#2,"nombre del nudo";nomb
$(n1):nomb$(n1)=UPPER$(nomb$(n1))
1090 RETURN
1100 :
1110 REM DIBUJA NUDOS
1120 :
1130 CLS
1140 FOR i=1 TO n1
1150 MOVE matriz(i,21)+15,matriz(i,
22)
1160 FOR j=0 TO 360 STEP 10
1170 DRAW matriz(i,21)+15*COS(j),ma
triz(i,22)+15*SIN(j)
1180 NEXT j
1190 MOVE matriz(i,21)+10,matriz(i,
22)
1200 FOR j=0 TO 360 STEP 10
1210 DRAW matriz(i,21)+10*COS(j),ma
triz(i,22)+10*SIN(j)
1220 NEXT j
1230 MOVER 2,0:IF matriz(i,28)=0 AN
D matriz(i,29)=0 THEN FILL 2 ELSE F
ILL 3
1240 MOVE matriz(i,21)-8,matriz(i,2
2)+8:TAG:PRINT CHR$(64+i);:TAGOFF
1250 NEXT i
1260 RETURN
1270 :
1280 REM POSICIONA NUDO
1290 :
1300 MOVE 0,400:TAG:PRINT CHR$(224)
;:posx=0:posy=400
1310 mov1=0:mov2=0
1320 IF INKEY(0)<>-1 THEN mov1=5:mo
v2=0 ELSE IF INKEY(2)<>-1 THEN mov1
=-5:mov2=0 ELSE IF INKEY(1)<>-1 THE
N mov1=0:mov2=5 ELSE IF INKEY(8)<>-
1 THEN mov1=0:mov2=-5 ELSE IF INKEY
(18)<>-1 THEN RETURN ELSE GOTO 1320
1330 AX=POSX:AY=POSY
1340 AX=AX+MOV2:AY=AY+MOV1
1350 IF AX>640-32 OR AX<0 THEN GOTO
1320
1360 IF AY>400 OR AY<64 THEN GOTO 1
320
1370 COND=0:REM si a la salida vale
1, ira a la 6020
1380 FOR I=1 TO n1
1390 AXY=SQR((AX-MATRIZ(I,21))^2+(A
Y-MATRIZ(I,22))^2)
1400 IF AXY<=96 THEN COND=1
1410 NEXT I
1420 IF COND=1 THEN 1340
1430 MOVE POSX,POSY:TAG:PRINT CHR$(
128);:TAGOFF
1440 POSX=AX:POSY=AY
1450 MOVE POSX,POSY:TAG:PRINT CHR$(
224);:TAGOFF
1460 GOTO 1320
1470 :
1480 REM INTRODUCIR ANTECEDENTE
1490 :
1500 NIVEL=1
1510 FOR K=1 TO N1-1
1520 MOVE MATRIZ(K,21)-8,MATRIZ(K
,22)+8:TAG:PRINT CHR$(224);:TAGOFF
1530 CLS#2:PRINT#2,"ES "jNOMB$(K);
" REQUISITO S/N":PRINT#2:PRINT#2,CHR
$(64+K);" es la tarea "jNOMB$(K);
1540 E$=UPPER$(INKEY$)
1550 WHILE E$<>"N" AND E$<>"S"
1560 E$=UPPER$(INKEY$)
1570 WEND
1580 IF E$="S" OR E$="N" THEN matri
z(n1,k)=1:nivel1=matriz(k,30)
1590 IF nivel1>nivel THEN nivel=niv
el1
1600 NEXT k
1610 matriz(n1,30)=nivel
1620 RETURN

```


GESTION

```

1630 :
1640 REM DIBUJA ANTECEDENTE
1650 :
1660 IF n1<2 THEN RETURN
1670 FOR I=1 TO N1
1680 J=1
1690 WHILE J<21
1700 WHILE MATRIZ(I,J)<>0
1710 AUX1=J
1720 AUX3=MATRIZ(I,21)-MATRIZ(AUX1,21)
1730 AUX2=MATRIZ(I,22)-MATRIZ(AUX1,22)
1740 IF AUX2=0 THEN ANGULO=0 ELSE A
NGULO=ATN(ABS(AUX3/AUX2))
1750 IF AUX2<=0 AND AUX3<0 THEN MOV
E MATRIZ(I,21)+15*COS(ANGULO),MATRIZ(I,22)+15*SIN(ANGULO):DRAW MATRIZ(AUX1,21)-15*SIN(ANGULO),MATRIZ(AUX1,22)-15*COS(ANGULO)
1760 IF AUX2<=0 AND AUX3>0 THEN MOV
E MATRIZ(I,21)+15*COS(ANGULO),MATRIZ(I,22)-15*SIN(ANGULO):DRAW MATRIZ(AUX1,21)-15*SIN(ANGULO),MATRIZ(AUX1,22)+15*COS(ANGULO)
1770 IF AUX2>0 AND AUX3<0 THEN MOV
E MATRIZ(I,21)-15*COS(ANGULO),MATRIZ(I,22)+15*SIN(ANGULO):DRAW MATRIZ(AUX1,21)+15*SIN(ANGULO),MATRIZ(AUX1,22)-15*COS(ANGULO)
1780 IF AUX2>0 AND AUX3>0 THEN MOV
E MATRIZ(I,21)-15*COS(ANGULO),MATRIZ(I,22)-15*SIN(ANGULO):DRAW MATRIZ(AUX1,21)+15*SIN(ANGULO),MATRIZ(AUX1,22)+15*COS(ANGULO)
1790 J=J+1
1800 WEND
1810 J=J+1:WEND
1820 NEXT I
1830 RETURN
1840 :
1850 REM CONTROL DE FIN
1860 :
1870 CLS#2:PRINT#2,"Meter mas nudos
puls N          menu pulsa
M"
1880 WHILE INKEY(46)=-1 AND INKEY(38)=-1
1890 WEND
1900 IF INKEY(46)<>-1 THEN GOSUB 2540
1910 IF INKEY(38)<>-1 THEN sal=1:RE
TURN
1920 :
1930 REM introducir comentario
1940 :
1950 CLS
1960 ERASE com$:DIM com$(200)
1970 CLS:PRINT"COMENTARIO PARA LA T
AREA "nomb$(n1):PRINT
1980 SAL=1:CO=1
1990 bs$=CHR$(127):e$=CHR$(13):RET$
=CHR$(8)
2000 WHILE SAL
2010 I$=INKEY$:IF I$="" THEN 2010
2020 IF I$=E$ THEN SAL=0:GOTO 2060
2030 IF I$=BS$ THEN IF CO=1 THEN PR
INT CHR$(7):GOTO 2060 ELSE CO=CO+1:
COM$(CO)="":PRINT RET$;PRINT " ";:
PRINT RET$;GOTO 2060
2040 PRINT I$;COM$(CO)=I$:CO=CO+1
2050 IF CO=200 THEN SAL=0
2060 WEND
2070 RETURN
2080 REM GRABAR DATOS
2090 :
2100 IF LEN(P$)<=8 THEN PR$=P$+".DE
C" ELSE PR$=LEFT(P$,8)+".DEC"
2110 CLS:LOCATE 10,10:PRINT"GRABAND
O : "PR$
2120 OPENOUT PR$
2130 PRINT#9,P$
2140 PRINT#9,N1
2150 FOR N=1 TO N1
2160 PRINT #9,NOMB$(N)
2170 PRINT#9,Come$(n)
2180 FOR K=1 TO 31
2190 PRINT CHR$(127);
2200 PRINT#9,MATRIZ(N,K)
2210 NEXT

```

```

2220 PRINT:NEXT
2230 CLOSEOUT
2240 RETURN
2250 :
2260 REM CARGAR DATOS
2270 :
2280 CLS:CAT
2290 LOCATE 10,20:PRINT"NOMBRE DEL
PROYECTO"
2300 LOCATE 16,22:INPUT " ",PR$
2310 IF INSTR(PR$,".")<>0 THEN 2300
2320 PR$=PR$+".DEC":OPENIN PR$
2330 INPUT #9,P$
2340 CLS:LOCATE INT(LEN(P$)/2),14:P
RINT P$
2350 LOCATE 11,10:PRINT"CARGANDO PR
OYECTO"
2360 INPUT #9,N1
2370 FOR N=1 TO N1
2380 PRINT CHR$(127);
2390 INPUT #9,NOMB$(N)
2400 INPUT#9,Come$(n)
2410 FOR K=1 TO 31
2420 PRINT CHR$(127);
2430 INPUT#9,MATRIZ(N,K)
2440 NEXT
2450 PRINT:NEXT
2460 CLOSEIN
2470 RETURN
2480 :
2490 REM nuevo proyecto
2500 :
2510 ERASE matriz,nomb$,com$,come$:
DIM matriz(20,31),nomb$(20),com$(20
0),come$(20)
2520 n1=0:CLS:LOCATE 1,10:INPUT "
NOMBRE DEL PROYECTO ",p$:p$=UPPER$
(p$)

```

LISTA DE VARIABLES

MATRIZ (I,J)	Guarda los antecedentes de cada uno de tal forma que el elemento I es antecedente del J si en esta posición hay almacenado un uno.
NOMBS (I)	Almacena el nombre de la tarea I
COM\$	Almacena, carácter a carácter el comentario de la tarea en curso.
COMES (I)	En esta matriz se almacena el contenido de COM\$ para su almacenamiento.
N1	Almacena el número de nudos introducidos.
MP,MPS	Opciones del menú principal.
MM,MMS	Opciones del menú de modificaciones.
MD,MDS	Opciones del menú de modificar datos.
MR,MRS	Opciones del menú de resultados.
MUS	Matriz transitoria para imprimir menús.
P\$	Nombre del proyecto.
SAL	Si es 0 vuelve al menú principal.
SAL1	Si es 0 abandona el programa.
CN	Opción elegida.

```

2530 GOSUB 1130
2540 GOSUB 1300:n1=n1+1:matriz(n1,2
1)=posx:matriz(n1,22)=posy
2550 GOSUB 1130:GOSUB 1060:GOSUB 15
00
2560 GOSUB 1950
2570 GOSUB 1130
2580 GOSUB 1660
2590 GOSUB 1870
2600 RETURN
2610 :
2620 REM introducir consecuentes
2630 :
2640 nivel=50
2650 FOR k=1 TO n1-1
2660 CLS#2:PRINT#2,"Es ";nomb$(k);"
CONSECUENTE s/n":PRINT#2:PRINT#2,C
HR$(64+k); " es la tarea ";nomb$(k)
2670 e$=UPPER$(INKEY$)
2680 WHILE e$<>"N" AND e$<>"S"
2690 e$=UPPER$(INKEY$)
2700 WEND
2710 IF e$="S" THEN matriz(k,n1)
2720 NEXT
2730 RETURN
2740 :
2750 REM nudo?
2760 :
2770 CLS:CLS #2:LOCATE 1,8:INPUT"no
mbre ha quitar";a$:a$=UPPER$(a$)
2780 num=0
2790 FOR i=1 TO n1
2800 IF a$=nomb$(i) THEN num=i:i=n1
2810 NEXT
2820 IF num=0 THEN CLS#2:PRINT#2,"N
O EXITE EL NUDO ";a$:GOSUB 2740:ELS
E RETURN
2830 RETURN
2840 :
2850 REM cambiar duracion
2860 :
2870 GOSUB 2740
2880 LOCATE 1,12:PRINT "el nudo ";n
omb$(num);" tiene duracion ";matriz
(num,31)
2890 LOCATE 1,14:PRINT "el nudo ";n
omb$(num):INPUT " debe tener duraci
on ",matriz(num,31)
2900 LOCATE 1,18:PRINT"el nudo ";no
mb$(num);" tiene duracion ";matriz(
num,31)
2910 GOSUB 4700:RETURN
2920 :
2930 REM modificar comentario
2940 :
2950 GOSUB 2740:CLS
2960 ERASE com$:DIM com$(200)
2970 PRINT come$(num)
2980 LOCATE 6,7:PRINT"modificando c
omentarío "nomb$(num)
2990 GOSUB 1980:sal=1
3000 RETURN
3010 :
3020 REM modificar antecedentes
3030 :
3040 GOSUB 2740:CLS
3050 LOCATE 11,3:PRINT"BORRO ANTECE
DENTES"
3060 FOR i=1 TO 20
3070 matriz(num,i)=0
3080 NEXT
3090 FOR i=1 TO 300:NEXT
3100 LOCATE 1,10:PRINT"CUANTOS ANTE
CEDENTES TIENE EL NUDO : ";nomb$(nu
m);:INPUT " ",c1
3110 IF c1>20 THEN 3100
3120 num1=num
3130 FOR i=1 TO c1
3140 GOSUB 2740
3150 matriz(num1,num)=1
3160 NEXT
3170 RETURN

```



```

3180 :
3190 REM cambiar nombre a un nudo
3200 :
3210 GOSUB 2740
3220 LOCATE 1,12:PRINT "el nudo ";n
omb$(num);" debe tener nombre";:INP
UT ":",nomb$(num):nomb$(num)=UPPER$(
nomb$(num))
3230 LOCATE 1,18:PRINT "el nudo ahor
a se llama ";nomb$(num)
3240 GOSUB 4700:RETURN
3250 :
3260 REM menu modificar
3270 :
3280 sal=1
3290 WHILE sal
3300 FOR n=0 TO mm
3310 mu$(n)=mm$(n)
3320 NEXT
3330 lin=9:fl=18:inc=3:buc=mm
3340 CLS
3350 GOSUB 3390
3360 ON cn GOSUB 3790,3540,3660,290
3370 WEND
3380 sal=1:RETURN
3390 :
3400 REM imprime menus
3410 :
3420 LOCATE #2,7,2:PRINT #2,"elige
opcion con el cursor
3430 l=LEN(mu$(0)):ps=lin
3440 LOCATE INT((40-l)/2),3:PRINT m
u$(0)
3450 FOR n=1 TO buc
3460 l=LEN(mu$(n))
3470 LOCATE INT((40-l)/2),ps:PRINT
mu$(n)
3480 ps=ps+inc
3490 NEXT
3500 ps=lin:cn=1
3510 GOSUB 5290
3520 RETURN
3530 :
3540 REM insertar
3550 :
3560 GOSUB 1130
3570 GOSUB 1300:n1=n1+1:matriz(n1,2
1)=posx:matriz(n1,22)=posy
3580 GOSUB 1130
3590 GOSUB 1060
3600 GOSUB 1500
3610 GOSUB 2620
3620 GOSUB 1950
3630 GOSUB 1130
3640 GOSUB 1660
3650 RETURN
3660 :
3670 REM modificar datos
3680 :
3690 FOR n=0 TO md
3700 mu$(N)=md$(N)
3710 NEXT
3720 lin=8:fl=16:inc=2:buc=md:sal=1
:cn=1
3730 WHILE sal
3740 CLS
3750 GOSUB 3390
3760 ON cn GOSUB 3180,2840,2920,301
0,290
3770 WEND
3780 sal=1:RETURN
3790 :
3800 REM QUITAR NUDO
3810 :
3820 GOSUB 2740
3830 FOR i=num TO n1
3840 FOR j=1 TO 31
3850 IF i<n1 THEN matriz(i,j)=matr
iz(i+1,j) ELSE matriz(i,j)=0
3860 NEXT
3870 IF i<n1 THEN nomb$(i)=nomb$(i+
1) ELSE nomb$(i)=""
3880 NEXT
3890 FOR j=1 TO 20
3900 FOR i=num TO n1
3910 IF i<n1 THEN matriz(j,i)=matr
iz(j,i+1) ELSE matriz(j,i)=0
3920 NEXT
3930 NEXT
3940 n1=n1-1
3950 LOCATE 1,18:PRINT "el nudo ";a$

```

```

;" ha sido borrado":FOR i=1 TO 300:
NEXT
3960 RETURN
3970 :
3980 REM menu resultados
3990 :
4000 FOR I=1 TO N1:MATRIZ(I,23)=0:M
ATRIZ(I,24)=0:MATRIZ(I,25)=0:MATRIZ
(I,26)=0:MATRIZ(I,27)=0:MATRIZ(I,28
)=0:MATRIZ(I,29)=0:NEXT
4010 CLS:CLS #2:LOCATE 3,12:PRINT "E
SPERE ESTOY CALCULANDO RESULTADOS"
4020 GOSUB 390
4030 FOR n=0 TO mr
4040 mu$(N)=mr$(N)
4050 NEXT
4060 lin=9:fl=18:inc=3:buc=mr:sal=1
4070 WHILE sal
4080 CLS
4090 GOSUB 3390
4100 ON cn GOSUB 4270,4140,4760,290
4110 WEND
4120 sal=1:RETURN
4130 DRAW matriz(i,21)+10*COS(j),ma
triz(i,22)+10*SIN(j)
4140 :
4150 REM camino critico
4160 :
4170 GOSUB 1130
4180 GOSUB 1660
4190 LOCATE#2,1,1:PRINT#2,"PULSA M
VOLVER AL MENU"
4200 e$=UPPER$(INKEY$)
4210 WHILE e$<>"M" AND n1>0
4220 FOR i=1 TO n1
4230 LOCATE#2,1,2:PRINT#2,CHR$(i+64
);" Es el nudo ";nomb$(i)
4240 FOR j=1 TO 300:NEXT j
4250 NEXT:e$=UPPER$(INKEY$)
4260 WEND:RETURN
4270 :
4280 REM visualizar resultados
4290 :
4300 CLS:LOCATE 7,10:PRINT "listamos
por impresora (s/n)
4310 i$=UPPER$(INKEY$):IF i$="" THE
N 4310
4320 IF i$="S" THEN impre=1 ELSE im
pre=0
4330 CLS:i$=STRING$(10,"#"):i1$="&
"
4340 IF impre THEN FOR k=1 TO 8:PRI
NT#8,CHR$(13):NEXT
4350 FOR n=1 TO n1
4360 LOCATE 4,2:PRINT "proyecto";STR
ING$(19,".");p$
4370 IF impre THEN PRINT #8,TAB(24)
;"proyecto";STRING$(19,".");p$
4380 LOCATE 4,3:PRINT "nudo";STRING$
(23,".");CHR$(64+n)
4390 IF impre THEN PRINT #8,TAB(24)
;"nudo";STRING$(23,".");CHR$(64+n)
4400 LOCATE 4,5:PRINT "nombre del nu
do";STRING$(12,".");nomb$(n)
4410 IF impre THEN PRINT #8,TAB(24)
;"nombre del nudo";STRING$(12,".");
nomb$(n)
4420 LOCATE 4,6:PRINT "comienzo";STR
ING$(19,".");matriz(n,23)
4430 IF impre THEN PRINT #8,TAB(24)
;"comienzo";STRING$(19,".");matriz(
n,23)
4440 LOCATE 4,7:PRINT "final";STRING
$(22,".");matriz(n,24)
4450 IF impre THEN PRINT #8,TAB(24)
;"final";STRING$(22,".");matriz(n,2
4)
4460 LOCATE 4,8:PRINT "duracion";STR
ING$(19,".");matriz(n,31)
4470 IF impre THEN PRINT #8,TAB(24)
;"duracion";STRING$(19,".");matriz(
n,31)
4480 LOCATE 4,10:PRINT "holgura inde
pendiente";STRING$(6,".");matriz(n,
29)
4490 IF impre THEN PRINT #8
4500 LOCATE 4,11:PRINT "holgura inte
rferente";STRING$(7,".");matriz(n,2
8)
4510 IF impre THEN PRINT #8,TAB(24)

```

```

;"holgura independiente";STRING$(6,
".");matriz(n,29)
4520 IF impre THEN PRINT #8,TAB(24)
;"holgura interferente";STRING$(7,"
.");matriz(n,28)
4530 LOCATE 4,12:PRINT "holgura tota
l";STRING$(14,".");matriz(n,27)
4540 IF impre THEN PRINT #8,TAB(24)
;"holgura total";STRING$(14,".");ma
triz(n,27)
4550 IF impre THEN PRINT #8:PRINT #
8:PRINT #8,SPC(41);"comentario ":"P
RINT #8:PRINT #8
4560 LOCATE 11,16:PRINT "comentario"
4570 c1$=MID$(come$(n),1,40)
4580 c2$=MID$(come$(n),41,40)
4590 c3$=MID$(come$(n),81,40)
4600 c4$=MID$(come$(n),121,40)
4610 c5$=MID$(come$(n),161,40)
4620 LOCATE 1,17:PRINT come$(n)
4630 IF impre THEN PRINT #8,TAB(20)
;c1$:PRINT#8,TAB(20);c2$:PRINT#8,TA
B(20);c3$:PRINT#8,TAB(20);c4$:PRINT
#8,TAB(20);c5
4640 IF impre THEN IF n MOD 2=0 THE
N FOR k=1 TO 16:PRINT#8,CHR$(13):NE
XT:GOTO 4660
4650 IF impre THEN FOR k=1 TO 16:PRI
NT#8,CHR$(13):NEXT
4660 GOSUB 4700
4670 CLS:NEXT
4680 IF impre THEN PRINT#8,CHR$(12)
4690 RETURN
4700 :
4710 REM pausa
4720 :
4730 CLS #2:LOCATE #2,5,2:PRINT#2,"
pulsar una tecla para continuar"
4740 i$=INKEY$:IF i$="" THEN 4740
4750 RETURN
4760 :
4770 REM DIAGRAMA DE BARRAS
4780 :
4790 CLS:LON=0
4800 LOCATE 1,1:PRINT "PROYECTO ";P$
4810 LOCATE 1,3:PRINT "NUDOS"
4820 LOCATE 1,4:PRINT "*****
*****"
4830 FOR I=1 TO N1
4840 LON1=MATRIZ(I,24)
4850 IF LON<LON1 THEN LON=LON1
4860 LOCATE 1,5+I:PRINT LEFT$(NOMB$
(I),20)
4870 LOCATE 20,5+I:PRINT "*"
4880 NEXT
4890 LON1=20/LON
4900 L1=LEN(STR$(LON))
4910 LOCATE 20,3:PRINT "0":LOCATE 40
-L1,3:PRINT LON
4920 FOR I=1 TO N1
4930 MOVE INT((20*16)+MATRIZ(I,23)*
16*LON1),336-16*I:DRAW 0,-15:DRAW
INT(MATRIZ(I,31)*16*LON1),0:DRAW
0,15:DRAW INT(-1*MATRIZ(I,31)*16*L
ON1),0:MOVER 4,-6:FILL 1
4940 MOVE INT((20*16)+MATRIZ(I,24)*
16*LON1),336-16*I:DRAW 0,-15:DRAW
INT(MATRIZ(I,28)*16*LON1),0:DRAW
0,15:DRAW INT(-1*MATRIZ(I,28)*16*L
ON1),0:IF MATRIZ(I,20)>0 THEN MOVER
4,-6:FILL 2
4950 MOVE INT((20*16)+MATRIZ(I,24)
+MATRIZ(I,28)*16*LON1),336-16*I:DR
AW 0,-15:DRAW INT(MATRIZ(I,29)*16
*LON1),0:DRAW 0,15:DRAW INT(-1*MA
TRIZ(I,29)*16*LON1),0:IF MATRIZ(I,2
9)>0 THEN MOVER 4,-6:FILL 3
4960 NEXT
4970 GOSUB 4700:RETURN
4980 FOR n=0 TO mp:READ mp$(n):NEXT
4990 FOR n=0 TO md:READ md$(n):NEXT
5000 FOR n=0 TO mr:READ mr$(n),mr$(
n):NEXT
5010 DATA "menu principal","salir",
"grabar","cargar","iniciar","modifi
car","resultados"
5020 DATA "modificar datos","nombre
","duracion","comentario","antecede
nte","volver al menu principal"
5030 DATA "menu modificar","menu re

```



```

sultados", "quitar nudo", "ver dato",
"insertar nudo", "camino critico", "m
odificar datos", "diagrama de barras
", "volver al menu principal", "vuelve
r al menu principal"
5040 :
5050 REM menu principal
5060 :
5070 sal=1
5080 WHILE sal=1
5090 FOR n=0 TO mp
5100 mu$(N)= mp$(n)
5110 NEXT
5120 lin=6:fl=16:inc=2:buc=mp
5130 CLS
5140 GOSUB 3390
5150 ON cn GOSUB 340,2080,2260,2490
,3250,3970
5160 WEND
5170 WHILE INKEY$("<")<":WEND
5180 LOCATE 8,4
5190 PRINT "si abandonas el programa
"
5200 LOCATE 8,6
5210 PRINT "perderas todos los datos
"
5220 LOCATE 9,8
5230 PRINT "cofirma pulsando -s-"
5240 LOCATE 2,10
5250 PRINT "cualquier otra tecla vue
lve al menu
5260 i$=UPPER$(INKEY$):IF i$="" THE
N 5260
5270 IF i$("<")<"S" THEN sal=1:GOTO 508
0
5280 sal=1:RETURN
5290 :
5300 REM mueve cursor por menu
5310 :
5320 PEN 3:LOCATE INT((40-LEN(mu$(c
n)))/2),ps:PRINT mu$(CN)
5330 WHILE INKEY(18)=1
5340 IF ps<lin THEN ps=fl:cn=buc
5350 IF ps>fl THEN ps=lin:cn=1
5360 LOCATE INT((40-LEN(mu$(cn)))/2
),ps:PRINT mu$(CN)
5370 IF INKEY(0)<)-1 THEN PEN 1:LOC
ATE INT((40-LEN(mu$(cn)))/2),ps:PRI
NT mu$(CN):ps=ps-inc:cn=cn-1:PEN 3
5380 IF INKEY(2)<)-1 THEN PEN 1:LOC
ATE INT((40-LEN(mu$(cn)))/2),ps:PRI
NT mu$(CN):ps=ps+inc:cn=cn+1:PEN 3
5390 WEND:PEN 1
5400 WHILE INKEY$("<")<":WEND
5410 RETURN
5420 CLS
5430 MODE 2
5440 WINDOW #2,1,80,23,24
5450 LOCATE 6,12:a$="EN EL PROGRAMA
APARECERA UN MENU PRINCIPAL CON LA
S DISTINTAS OPCIONES.
EXAMINEMOS CADA UNA DE E
LLAS"
5460 GOSUB 5640:CLS
5470 LOCATE 27,1:a$="OPCION INICIAL
IZAR":GOSUB 5640:PRINT
5480 a$="EN ESTA OPCION SE INTRODUC
EN LOS DATOS PARA HALLAR EL CAMINO
CRITICO.EL ORDENA -DOR NOS PREGUNTA
RA POR EL NOMBRE DE CADA TAREA(NUDO
) Y SU DURACION.TAMBIEN NOS DIRA S
I LOS NUDOS ANTERIORES SON ANTECE
DENTES DEL NUDO EN QUE ESTAMOS. ":GOS
UB 5640
5490 PRINT:a$="RECUERDA QUE SOLO SO
N ANTECEDENTES LOS ULTIMOS NUDOS D
E LOS QUE DEPENDA. AUNQUEESTOS ULTI
MOS DEPENDAN DE ALGUN PRIMERO,CUAND
O EL ORDENADOR NOS PREGUNTE UNO POR
UNO SI SON ANTECEDENTES SOLO CONSID
ERAREMOS TALES LOS QUE SE UNEN DIRE
CTAMENTE"
5500 GOSUB 5640:PRINT
5510 a$="APARECERA EN LA PANTALLA
UN SIMBOLO QUE PUEDES MOVER CON LAS
TECLAS DEL CURSOR CUANDO PULSES [R
ETURN] EL ORDENADOR DIBUJARA UN NUD
O":GOSUB 5640:PRINT
5520 a$="SI VES QUE EL MUJECO SE PA
RA NO TE IMPACIENTES. ESTA CALCULA

```

```

NDO DONDE SE PUEDE PONER SIN ESTROP
EAR EL DIBUJO.CUANDO DE EL SALTO PU
EDES CONTINUAR.":GOSUB 5640:PRINT
5530 a$="POR ULTIMO, PUEDES ASIGNAR
UN COMENTARIO A CADA TAREA SI LO D
ESEAS.":GOSUB 5640
5540 CLS:LOCATE 30,1:a$="OPCION MOD
IFICAR":GOSUB 5640:PRINT
5550 a$="CON ESTA OPCION PUEDES MOD
IFICAR CUALQUIERA DE LOS DATOS QUE
HAS INTRODUCIDO.":GOSUB 5640:PRINT
5560 a$="POSEES UN COMPLETO SUBMENU
DE MODIFICACION PARA ELEGIR LA OPC
ION.":GOSUB 5640:PRINT
5570 LOCATE 37,10:a$="GRABAR":GOSUB
5640:PRINT:a$="ESTA OPCION GUARDA
EN DISCO TODOS LOS DATOS DEL PROYE
CTO PARA CARGARLOS EN EL MOMENTO Q
UE SE DESEE SIN TENER QUE RECLEARLO
S DE NUEVO.EL ORDENADOR LOS SALVA
CON EL NOMBRE DEL PROYECTO Y LA EX
TENSION.DE
5580 GOSUB 5640:CLS
5590 LOCATE 34,1:a$="RESULTADOS":GO
SUB 5640:PRINT
5600 a$="EL SUBMENU PERMITE ELEGIR
ENTRE VER DATOS,DIAGRAMA DE BARRA
S O CAMINO CRITICO.SI PULSAS VER
DATOS SALDRAN EN PANTALLA CADA UNO
DE LOS DATOS CALCULADOS PARA LOS DI
FERENTES NUDOS.":GOSUB 5640:PRINT
5610 a$="EL DIAGRAMA DE BARRAS TE
MUESTRA GRAFICAMENTE LAS TAREAS QUE
NO PERTENECEN AL CAMINO CRITICO
CON LA HOLGURA DE CADA UNA DE ELL
AS REPRESENTADA EN ROJO.TODASLAS TA
REAS TOTALMENTE AMARILLAS SON CRIT
ICAS.":GOSUB 5640:PRINT
5620 a$="POR ULTIMO SI ELIGES VER
EL CAMINO CRITICO,TE APARECERA DE
NUEVO LA RED QUE TUNAS CREADO CON L
OS NUDOS DEL CAMINO CRITICO EN AZUL
Y LOS QUE NO PERTENECEN A ELEN ROJ
O.":GOSUB 5640
5630 GOTO 5710:CLS
5640 FOR N=1 TO LEN(a$)
5650 Z=INT(RND*500):V=INT(RND*11+5)
: SOUND 1,2,1,V
5660 PRINT MID$(a$,N,1);:NEXT N:PRI
NT " "
5670 CLS#2:LOCATE #2,19,24:PRINT #2
,"PULSE CUALQUIER TECLA PARA CONTIN
UAR"
5680 E$=INKEY$
5690 IF INKEY$="" THEN 5690
5700 RETURN
5710 CLS:CLS#2
5720 LOCATE 15,12:PRINT"VAMOS A VER
UN EJEMPLO DE COMO HACER UNA RED":
a$="":GOSUB 5640:CLS
5730 a$="SUPONGAMOS QUE ES USTED EL
PRODUCTOR DE UNA PELICULA. USTED D
ESEEA VERLA TERMINA DA CUANTO ANTES
PUES CADA DIA QUE PASA LE CUESTA DI
NERO":GOSUB 5640
5740 PRINT
5750 a$="SUPONGA QUE LAS TAREAS QUE
USTED TIENE QUE REALIZAR PARA COMP
LETAR LA PELICULA SON ESTAS ":GOS
UB 5640
5760 PRINT:a$="0-CONTRATAR LOS MEDI
OS 1-RODAR EXTERIORES 2-RODAR INTE
RIORES 3-RODAR EF
ECTOS ESPECIALES 4- EMPALMAR 5-BAN
DA SONORA 6-TITULOS
7-DISTRIBUCION DE LA PELICULA":GOSU
B 5640
5770 PRINT:a$="SUPONGAMOS QUE NO P
ODEMOS RODAR LOS EFECTOS ESPECIALES
(ND3) SI ANTES NO HEMOSACABADO D
E RODAR INTERIORES Y EXTERIORES.ES
TO SIGNIFICA QUE EL NUDO 1 Y EL 2
SON ANTECEDENTES DEL TRES":GOSUB 56
40
5780 PRINT:a$="EN ESTE CASO LA TARE
A DE COMIENZO ES LA 0. DE ELLA SALD
RAN DOS RAMAS DIFERENTES:RODAR INTE
RIORES Y EXTERIORES.LUEGO SI EL PRO
GRAMA FUERA REAL CUANDO ENTRASEMOS
EL NUDO 1(REPRESENTADO EN NUESTRO E
JEMPLO POR ND1) Y NOS PREGUNTASE ":

```

GESTION

```

GOSUB 5640
5790 LOCATE 69,17:a$="SI EL NUDO 0E
S ANTECEDENTE DIRIAMOS QUE SI. LO M
ISMO OCURRIRIA CON LA TAREA 2 (ND2)
,QUE TAM-BIEN TIENE POR ANTECEDENTE
EL NUDO 0.":GOSUB 5640:PRINT
5800 CLS
5810 a$="CUANDO NOS PREGUNTASE SI E
L NUDO 0 ES ANTECEDENTE DEL NUDO TR
ES DIRIAMOS QUE NO PERO SI LO ES DE
L NUDO 1(ND1,RODAR EXT, Y DEL ND2)
DE LO CUAL INFORMARIAMOS AL PROGRA
MA":GOSUB 5640
5820 PRINT:a$="POR ULTIMO IRIAMOS P
ONIENDO COMO ANTECEDENTES A PARTIR
DEL NUDO 4 AL NUDO INME-DIATAMENTE
ANTERIOR. ASI HASTA EL NUDO FINAL(
EL 7,DISTRIBUCION DE LA PELICULA).":
GOSUB 5640
5830 CLS:a$="VEAMOS AHORA COMO SER
IA EL GRAFO DE TODA LA OPERACION ":
GOSUB 5640
5840 CLS#2
5850 TAG
5860 PLOT 70,180:DRAW 70,220:DRAW 3
0,220:DRAW 30,180:DRAW 70,180:DRAW
70,200
5870 FOR N=1 TO 12:FOR G=1 TO 150:N
EXT G:MOVE 20,240:PRINT " ";:
MOVE 38,208:PRINT " ";:FOR G=1 TO
150:NEXT G:MOVE 38,208:PRINT"ND0";:
MOVE 20,240:PRINT"COMIENZO";:NEXT N
5880 MOVE 70,200:DRAW 138,100
5890 FOR N=1 TO 15:FOR G=1 TO 150:N
EXT G:MOVE 140,100:PRINT " ";:MOV
E 100,50:PRINT " ";:FOR G=1
TO 150:NEXT G:MOVE 140,100:PRINT"ND
2";:MOVE 100,50:PRINT"RODAR INT";:N
EXT N
5900 MOVE 125,112:DRAW 125,80:DRAW
166,80:DRAW 166,112:DRAW 125,112
5910 MOVE 70,200:DRAW 320,200:MOVE
166,100:DRAW 320,200
5920 FOR N=1 TO 15:FOR G=1 TO 150:N
EXT G:MOVE 100,230:PRINT " ";:
MOVE 203,208:PRINT " ";:FOR G=1 T
O 150:NEXT G:MOVE 100,230:PRINT"ROD
AR EXT";:MOVE 203,208:PRINT"ND1";:N
EXT N
5930 MOVE 200,215:DRAW 200,193:DRAW
230,193:DRAW 230,215:DRAW 200,215
5940 MOVE 230,200:DRAW 320,200:FOR
N=1 TO 15:FOR G=1 TO 150:NEXT G:MOV
E 322,208:PRINT " ";:MOVE 232,218:
PRINT " ";:FOR G=1 TO 150:N
EXT G:MOVE 322,208:PRINT "ND3";:MOV
E 232,218:PRINT"RODAR EFEC";:NEXT N
5950 MOVE 320,215:DRAW 320,193:DRAW
350,193:DRAW 350,215:DRAW 320,215
5960 MOVE 350,200:DRAW 600,200
5970 MOVE 375,218:PRINT"LOS DEMAS N
UDOS ESTAN";
5980 MOVE 375,190:PRINT"TODO EN ES
TA LINEA";
5990 TAGOFF:a$=" ":GOSUB 5640
6000 CLS:CLS#2:LOCATE 1,12:PRINT "P
ARA VER DE NUEVO EL EJEMPLO ENTRA
E-,-I- PARA INSTRUCCIONES,-C- PARA
COMENZAR"
6010 a$=UPPER$(INKEY$):IF a$="" THE
N 6010
6020 IF a$="E" THEN GOTO 5710
6030 IF a$="I" THEN GOTO 5420
6040 RETURN

```



P ara que tus dedos
 no realicen el trabajo duro, M.H. ANS
 TRAD lo hace por ti. Todos los listados que incluyen
 este logotipo se encuentran a tu disposición en un cas-
 sette mensual, solicítalo.

LA VERSATILIDAD DEL LOGO

Logo es un lenguaje infinitamente flexible y versátil, que permite abordar los problemas desde un punto de vista lógico y bastante cercano a la forma de razonar de las personas. Estas características se traducen en programas elegantes y extremadamente cortos, pero con mucha «miga», como el ejemplo que presenta Análogo esta semana.

José María Díaz



Ya hemos comentado en más de una ocasión que el Logo sirve para algo más que crear excelentes gráficos de tortuga. También es sabido que, como lenguaje pedagógico, hoy por hoy, no tienen rival. Pero, ¿cómo se comportaría el lenguaje si quisiéramos hacer una aplicación que no contempla nada de esto como, por ejemplo, la Lotería Primitiva?

Un programa de la «Loto», en Basic, no tiene ningún problema: es sencillo de hacer, y corto. Al fin y al cabo, sólo se trata de extraer 6 números al azar de 49 sin repetición.

El programa en Logo sólo usa dos procedimientos, uno de los cuales se ha creado en beneficio de los usuarios del 464/664, cuyo Logo (el 2.2), por razones de memoria disponible, es menos potente. Los poseedores del 6128 que no se preocupen. Ese procedimiento («memberp»), simplemente ignórenlo y usen la primitiva correspondiente del propio lenguaje, que, por supuesto, se llama igual. Deberá funcionar en todos los casos.

Vamos a analizar un poco la lógica del programa. hay que cumplir los siguientes objetivos:

- 1) Generar un número aleatorio entre 1 y 49.
- 2) Comprobar que no está repetido.
- 3) Si no lo está, agregarlo a una lista que al final dará los 6 números definidos.
- 4) Si está repetido, ir al paso 1.
- 5) Si la lista en esta etapa tiene 6 números, indica que son distintos, y el programa termina, proporcionando en pantalla el resultado.
- 6) Si no se han alcanzado los 6 números, repetir todo el ciclo e ir al paso 1.

Todo este asunto se encarga de gestionarlo el procedimiento «**LOTO**». Veamos cómo lo hace, estudiándolo línea a línea.

TO LOTO: LISTA

Definición del procedimiento que, por razones que se harán claras luego, posee un argumento, «**LISTA**», el cual representa, efectivamente, una lista Logo, es decir, una serie correlativa de objetos que forman un grupo para poder ser manipulado de distintas formas.

(LOCAL "NUM)

Esta línea es más una cuestión de estilo que de necesidad. La variable «num» no va a ser necesaria fuera del procedimiento «**LOTO**», por lo que la declaramos LOCAL A EL. La ventaja de este enfoque estriba en que nos aseguramos taxativamente



PROGRAMA NO RECURSIVO

```
to memberp :val :lista
if :lista = [] [op 1 = 0]
if :val = first :lista [op 1 = 1]
[op memberp :val bf :lista]
end

to loto :lista
(local "num)
pr [LOTERIA PRIMITIVA]
label "loop
make "num (1 + random 49)
if memberp :num :lista [pr :num pr [existe]]
[make "lista (se :num :lista)]
pr :lista
if count :lista = 6 [op 1 = 1]
go "loop
end
```

La razón de usar la expresión `1 = 1`, que es TRUE siempre, es también una costumbre, en mi opinión buena, que permite usar cualquier procedimiento, por complejo que sea, en una expresión condicional (es el truco usado en MEMBERP). No obstante, se podría haber puesto otro cosa, como `[OP "FIN DEL PROGRAMA]`, por ejemplo.

GO "LOOP

Equivalente del NEXT en Basic. Si en la sentencia anterior el programa no ha acabado, quiere decir que todavía no tenemos 6 números distintos, por lo que el procedimiento «salta» a la etiqueta "LOOP y repite todo el proceso.

VERSION RECURSIVA

```
to memberp :val :lista
if :lista = [] [op 1 = 0]
if :val = first :lista [op 1 = 1]
[op memberp :val bf :lista]
end

to loto :lista
(local "num)
make "num (1 + random 49)
if memberp :num :lista [pr :num pr [existe]]
[make "lista (se :num :lista)]
pr :lista
if count :lista = 6 [op 1 = 1] [op loto :lista]
end

to primitiva :lista
pr [LOTERIA PRIMITIVA]
op loto :lista
end
```

Quisiera resaltar aquí un fallo, intencionado, que puede causar problemas en casos más complejos: lo que he dicho en el párrafo está IMPLICITO, no se deduce a primera vista de la inspección del programa. Yo lo sé porque lo he escrito, y usted porque lo lee, pero hubiera sido más correcto incluir esto en la parte «ELSE» del condicional y escribir:

IF COUNT :LISTA = 6 [OP 1 = 1] [GO "LOOP]

Aquí se ve EXPLICITAMENTE que, si la condición no se cumple, el programa lo intenta de nuevo saltando a LOOP.

END

Creo que es autoexplicativo: se acabó.

Le toca el turno ahora al procedimiento MEMBERP.

Antes de estudiarlo línea a línea, vamos a ver un poco qué es lo que se supone que debe hacer.

Nuestro objetivo es averiguar si un objeto se encuentra presente en una lista o no: para ello, una forma razonable de abordarlo es comparar el «objeto» (nuestro número aleatorio) con el primer elemento de la lista; si son iguales, MEMBERP debe terminar y comunicar un resultado positivo; si son distintos, el proceso debe repetirse hasta que se llegue al fin de la lista o se encuentre una igualdad.

A mí, todo este farrago me suena a dos cosas:

1) Nos interesa, más que un resultado numérico, como 3, por ejemplo, un resultado LOGICO, cierto o falso, está o no está.

2) Como hay que mirar toda una lista, elemento por elemento, y en principio no sabemos la longitud de la misma, debe funcionar de forma RECURSIVA, repitiendo, pero SIN EL PRIMER ELEMENTO DE LA LISTA.

Basándonos en esto, ya podemos intentar «hincarle el diente» a MEMBERP.

TO MEMBERP :VAL :LISTA

Todo normal. Nuestro procedimiento tiene dos argumentos: el valor a comparar y la lista donde queremos saber si está o no.

IF :LISTA = [] [OP 1 = 0]

Por lógica, algo recursivo debe poseer una condición que, al cumplirse, cause la salida del procedi-

miento; de lo contrario, se repetirá indefinidamente. El problema es que la condición, su significado, suele estar IMPLICITO. En efecto, si se llega a un momento en que :LISTA está vacía ([]), quiere decir que el valor que buscamos no está, porque si estuviera, nunca habríamos llegado a la situación de lista vacía. Es una especie de razonamiento al revés y, para entenderlo al 100 por 100, conviene mirar el procedimiento desde el final, cuando se llama a sí mismo recursivamente.

La parte `[OP 1 = 0]`, UNA VEZ MAS, PROPORCIONA UN RESULTADO LOGICO DE FALSO, lo que se detecta como resultado en el procedimiento principal LOTO, como explicamos antes.

A1) IF :VAL = FIRST :LISTA

B1) [OP 1 = 1]

C1) [OP MEMBERP :VAL BF :LISTA]

En A1 indagamos si nuestro número (VAL) es igual al primer (FIRST) elemento de LISTA. Si es así, el procedimiento termina dando el resultado TRUE (`[OP 1 = 1]`), y LOTO «toma nota» de ello.

Si no se cumple la igualdad, se ejecuta C1, en la cual se llama de nuevo a MEMBERP con VAL pero con LISTA sin su primer elemento (BF, BUTFIRST, sin primero). Es decir, a la siguiente vuelta, el número de objetos de LISTA ha decrecido en una unidad, por lo que, tarde o temprano, se alcanzará la situación de igualdad o de lista vacía, asegurándonos de que MEMBERP terminará alguna vez. En C1, OP es necesario para que el resultado de las llamadas recursivas «sea notificado» al procedimiento llamador, a LOTO.

END

Sin comentarios.

Bueno, pues esto es todo. Parece mentira que de un programa tan corto se pueda sacar tanto jugo, pero Logo es así: de muy alto nivel, y lleno de riqueza y contenido oculto.

Nota: como regalo, también incluimos la versión recursiva del programa, con un procedimiento más de llamada. A todo esto, el programa no recursivo se arranca con:

loto []

y el recursivo con:

primitiva []

TU PROGRAMA DE RADIO

claro!



Audio 52

- Entrevistas a fondo
- Exitos en Soft
- Noticias en Hard
- Concursos

Programátelo: Sábados tarde de 5 a 7 horas.
En directo y con tu participación.

LA COPE A TOPE.

— RADIO POPULAR 54 EMISORAS O.M. —

En Barcelona Radio Miramar



WEST BANK

El West Bank es la entidad financiera más famosa de todo el Oeste; por sus puertas puede entrar cualquier cliente o un forajido sin escrúpulos, tu pistola es la que decide.

Todo comenzó el día en que al gran magnate de las finanzas Peter Money, se le ocurrió abrir una sucursal de su banco en pleno Oeste.

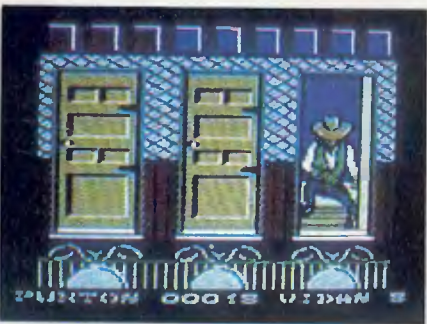
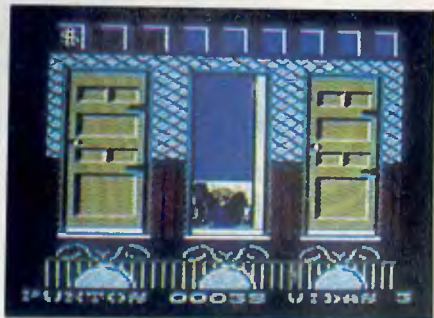
Con la llegada de la fiebre del oro, los pequeños poblados cercanos a las concesiones, eran puntos donde el polvo de oro y el dinero corrían a raudales, echándose en falta un sitio donde depositar las importantes sumas de dinero que algunos conseguían.

la West Bank Corporation, éste era que un banco en un poblado del Oeste, no es lo mismo que una sucursal en Wall Street.

En el Oeste hay una forma especial de tratar con la clientela, lo que en cualquier parte debe ser cordialidad y amabilidad, aquí se puede transformar en un asunto que se resuelve pistola en mano.

La sucursal va a ser visitada por una diversa variedad de clientes que quieren ingresar su dinero en sitio seguro, y otros que lo único que intentan es sacar dinero a punta de pistola en vez de utilizar su tarjeta de crédito.

La gran proliferación de delincuentes que ha traído la llegada del oro,



Esto, que no se escapó a la clara mente de Peter Money, fue el motivo que le decidió a comenzar su aventura bancaria en la salvaje tierra del oro.

Sólo un pequeño detalle perturba las previsiones del presidente de

hace que cualquier banco peligre seriamente ante tanto desalmado.

Para evitar que el banco sea visitado por gente indeseable, debe colocarse al frente de la sucursal a un hombre duro y sin escrúpulos que sepa ocuparse de la carne de presidio.

Este debe ocuparse de mantener a raya a toda esa marabunta y garantizar la seguridad del banco, por lo cual debe ser un experto pistolero desconectado totalmente del mundo de la banca.

Elegida la plantilla adecuada, se inaugura la sucursal y comienzan nuestros problemas.

Peter Money nos ha encomendado la custodia de su banco, nuestro revólver tiene que decidir quién viene a depositar dinero o a morir intentando apoderarse de los fondos del banco.

Esto, que en condiciones normales sería una tarea fácil, se convierte en una que requiere toda nuestra atención, debido a que los forajidos se





dos por los disparos de un forajido.

Entre los visitantes de la sucursal, tenemos a los típicos clientes que vienen con sendas bolsas de oro en las manos, y el característico maleante que empuñando un pistolón nos apunta con cara de pocos amigos.

Pero para dar más aliciente al juego, hay otros personajes que tan pronto pueden depositar dinero como darnos un tiro, con lo cual nuestra reacción debe ser rápida y precisa exigiéndonos una gran dosis de concentración.

Entre ellos tenemos al jugador de cartas, que enfundado en un elegante traje porta en su mano izquierda una bolsa de oro, con él hay que extremar el cuidado porque en cualquier momento puede desenfundar.

Otro es un mejicano que se planta en la puerta de nuestro banco con las



mezclan con la clientela haciendo difícil su identificación.

Nos encontramos en la sala de recepción, tras la ventanilla de ingresos, a nuestro frente se encuentran tres puertas por las que afluyen los parroquianos, cada vez que una de éstas se abre una sorpresa nos aguarda.

Si detrás de ella aparece un atracador, debemos hacer fuego inmediatamente antes de que nos dispare, pero por el contrario, si se trata de uno de nuestros distinguidos clientes le recibiremos con los brazos abiertos.

Por cada cliente que eliminemos por error, perderemos una vida, igual que cuando seamos alcanza-

manos listas para desenfundar, si lo hace debemos abrir fuego contra él, si no, dejarle marchar; disparar contra él sin que halla desenfundado nos hará perder una vida.

El otro tipo curioso es el enano de los sombreros, su grotesca figura aparece tras la puerta, coronada por cinco sombreros encajados uno sobre otro.

Si queremos saber lo que guarda bajo ellos, se los quitaremos a tiro limpio, apareciendo bajo el último su aportación al banco. Esta puede ser una bolsa de oro a la cual hemos de disparar para conseguirla, o una bomba que de ser alcanzada por nuestras balas, explotará, restándonos una vida más.

Mister JOYSTICK



Nuestra misión tras la barra, es que se deposite oro en las nueve ventanillas que constituyen la sucursal, cuando todas estén llenas habremos superado la fase y nos encontramos en la calle ante tres forajidos.

En el duelo hemos de hacer uso de toda nuestra sangre fría y esperar a que desenfunde uno de ellos para abrir fuego, y de esta forma liquidar a los tres obteniendo una vida extra, disparar sobre uno que no haya desenfundado no nos dará bonificación.

Nos encontramos ante un programa de producción nacional basado en su homónimo para las máquinas de billares, en el cual todo el interés está en ver qué nos espera detrás de cada puerta y actuar en consecuencia.

Un juego que requiere de toda nuestra concentración, la cual se encuentra repartida entre las tres puertas a vigilar, y en el que sólo nuestros buenos reflejos nos llevarán a las más altas puntuaciones.



GESTION DE CARGA DE FICHEROS

por J. J. Martín

Hasta hace bien poco, el hacer buenas aplicaciones era un reto para todo programador que se preciara, sin embargo, de un tiempo a esta parte las cosas han empezado a cambiar. No, no es que ya no se diseñe buen software, se hace, claro que se hace. Pero ahora una nueva voz se alza entre los usuarios: ¡No a los programas huraños!



tienen razón. ¿De qué sirve un programa que tarda cero coma cero cientos segundos en clasificar por orden de viudedad la lista de parados, si para ello hay que pulsar **CTRL + SHIFT + J** y a la vez teclear en el Keypad el código-alfanumérico-de-acceso-por-clave-de-entrada-mixta-para-personas-físicas-en-situación-no-activa? A la hora de diseñar un programa pueden olvidarse muchas cosas, pero hay una fundamental: va a ser utilizado por un humano que a lo sumo y en el mejor de los casos tiene dos manos, una cabeza y seguramente un jefe que le agobia pidiéndole los albaranes de antes de ayer. ¿Por qué no tener esto en cuenta a la hora de realizar un programa y facilitarle la vida al pobre usuario que, sin comerlo ni beberlo, se encuentra constantemente amenazado ante un tubo de rayos catódicos? ¿Y si encima queda bonito...? ¡Pues mejor!

La rutina que proponemos esta semana, de propósito muy general y utilizable en cualquier programa donde intervenga una gestión de ficheros o programas, se acerca a esta filosofía; lástima de «ratón».

Veamos con detalle su diseño:

10000 Rem de costumbre.

10010 Definimos un nuevo carácter, el 255, formado por líneas blancas (los ceros), y las líneas negras (los 255).

10020 Asignamos a la variable **BARRAS** una serie de caracteres para posteriormente escribirlos en pantalla.

10030 Ponemos la pantalla en modo 2.

10040-10050 Asignamos a la variable **VALIDAS** el contenido de aquellas teclas a las que la rutina va a responder, estos caracteres están contenidos en las siguientes teclas:

- 13: Return.
- 240: Cursor arriba.
- 241: Cursor abajo.
- 242: Cursor izquierda.
- 243: Cursor derecha.

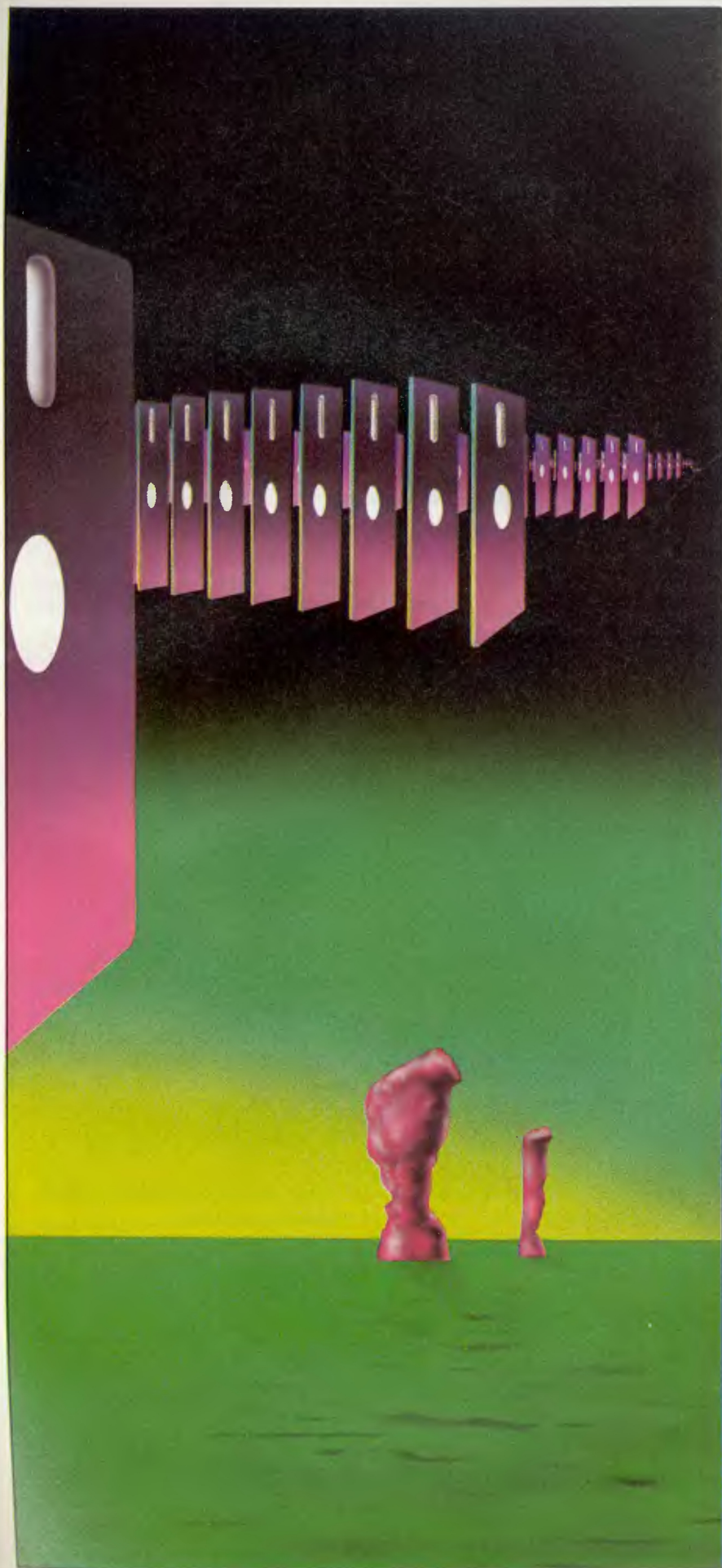
10060-10150 Ponemos el directorio del disco en pantalla e inicializamos algunas variables, que posteriormente utilizaremos. Las variables **FID** y **COD**, llevarán las coordena-

das de la fila y la columna del cursor. El poner la tinta 1 con el color 0, esto es el negro, nos sirve para que no se vea la impresión de la pantalla, y mostrarla al final de forma instantánea. En la línea 10090 hacemos el «**DIR**» del disco. El hecho de utilizar «**DIR**» y no el archiconocido «**CAT**» tiene una explicación que conviene aclarar.

El comando **CAT**, al contrario que **IDIR**, no acepta parámetros, lo que impide visualizar de forma selectiva una parte del directorio. Para utilizar correctamente esta sentencia, venida de CPM hasta el **AMSDOS** de nuestro **Amstrad**, sólo hace falta tener bien claro dos cosas, un asterisco (*) se traduce por cualquier palabra y una interrogación por cualquier letra. De este modo tecleando por ejemplo **IDIR "*" .BAS**, obtendremos en pantalla todos los programas Basic, grabados en el disco, y con **IDIR "T??.TXT"**, todos los ficheros con extensión **TXT** que empiecen por **T** y tengan una, dos o tres letras como nombre.

Volviendo al programa que nos ocupa, esta posibilidad, nos permite ofrecer al usuario como opción posible, sólo los ficheros que a nosotros nos interesen bien sea controlándolos por su nombre o por su extensión.

Una vez que hemos imprimido el directorio en pantalla, miramos me-



diente la instrucción VPOS en qué línea de la pantalla nos encontramos. Como al hacer el IDIR, se produce un salto de dos líneas, el valor de VPOS será tres unidades mayor que el número de líneas que ocupa el directorio real. Restamos estas tres líneas a VPOS y asignamos el contenido en TOPFIL, que a partir de este momento contiene el número de líneas del directorio.

En la línea 10010 imprimimos la variable BARRAS. Esta la utilizaremos para abortar la carga de un programa o fichero.

En la línea 10020 invertimos los colores de la pluma y del papel, para imprimir (línea 10130) la cabecera de la pantalla con un pequeño mensaje informativo.

Después restauramos los colores de la tinta y del papel, y enviamos la ejecución desde la línea 10150 a la línea 10260, para que se imprima el cursor por primera vez.

10160-10200 Chequeamos el teclado para ver si se ha pulsado alguna de las teclas del cursor o el Return. Para esto cargamos la variable TEC\$ con un asterisco, y después entramos en un bucle (Líneas 10180-10200), del que no saldremos hasta que no pulsemos una de las mencionadas teclas. El truco está en ver si la tecla pulsada está incluida (INSTR) dentro de la variable VALIDAS, que como recordará contenía las teclas a las que hay que «hacer caso». Como el "", esto es el vacío, está contenido en cualquier variable, en el caso de que no pulsemos ninguna tecla, asignamos a TEC\$ nuevamente el valor "".

10210-10290 Una vez la tecla ha sido reconocida como válida, pasamos a la rutina de movimiento del cursor o aceptación del fichero escogido (tecla Return). Si la tecla pulsada es Return, es decir la que contiene el "CHR\$(13)", el fichero seleccionado, donde se encuentra el cursor, es el que queremos cargar.

En el caso de que hallamos decidido mover el cursor, reimprimimos el nombre del fichero, recordemos que se encontraba en vídeo inverso, con pluma blanca y papel negro. Después variamos el valor del cursor conforme a la tecla del cursor pulsada, lo que realizamos en las líneas

10240 y 10250, tras lo que nos situamos en la correspondiente posición de la pantalla. Mediante la línea 10250 leemos el nombre del fichero sobre el que nos encontramos. Esto lo conseguimos leyendo once caracteres de la pantalla mediante la instrucción COPYCHR\$, caracteres que vamos almacenando en la variable FICHERO\$.

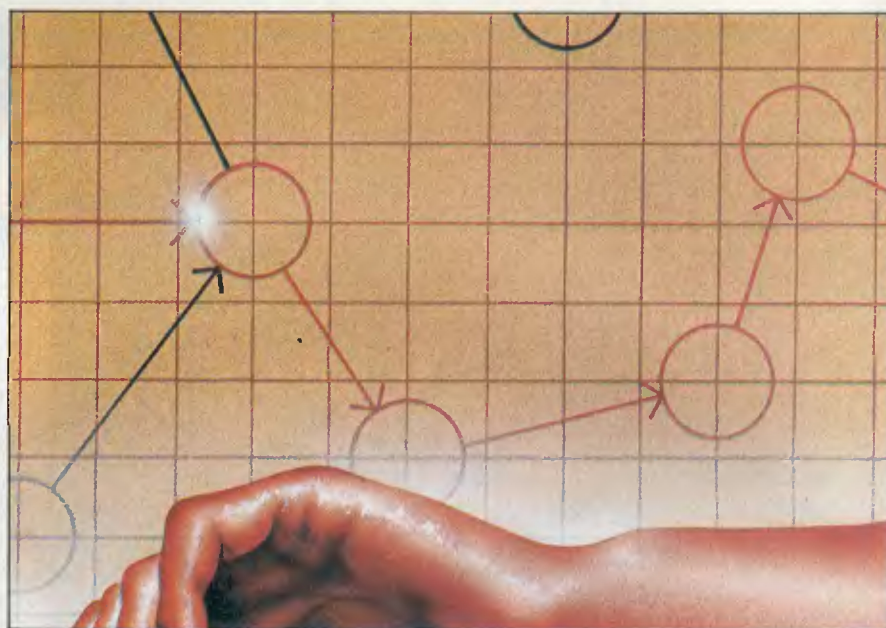
Una vez que hemos «capturado» el nombre del fichero sobre el que nos encontramos, retrocedemos once caracteres para imprimirlo en el mismo sitio, pero en video inverso para resaltarlo.

Una vez que hemos movido el cursor volvemos a la rutina de lectura del teclado en espera de que pulsemos una nueva tecla.

Recordemos que en el caso de que hubiésemos pulsado la tecla RETURN, y estamos en los finales de la rutina, el flujo del programa era enviado a la línea 10300.

10300-10320 Si nos encontramos en la barra de cancelación cargamos en la variable FICHERO\$, la palabra CANCELADO, lo que posteriormente controlaremos desde el programa principal. Por fin, devolvemos la ejecución al programa principal.

Como ejemplo de la utilización de esta rutina hemos realizado un pro-



grama muy sencillo. Nos muestra el directorio, nos da la opción de elegir el programa y lo ejecuta. Dada su sencillez no hay comentarios.

En resumen la moraleja de esta rutina es muy simple, las cosas bien hechas si «amables» dos veces bien hechas.



```

10000 REM Rutina de gestion de ficheros y programas
10010 SYMBOL 255,0,255,0,255,0,255,0,255
10020 barra$=STRING$(31,255)+"CANCELAR"+STRING$(41,255)
10030 MODE 2
10040 REM ---TECLAS VALIDAS
10050 valida$=CHR$(13)+CHR$(240)+CHR$(241)+CHR$(242)+CHR$(243)+CHR$(244)
10060 REM ---VISUALIZACION DE DIRECTORIO Y CARGA DE VARIABLES
10070 fid=4:cod=1
10080 INK 1,0:
10090 :DIR, "*,*"
10100 TOPFIL=VPOS(0)-3:LOCATE 1,TOPFIL+1:PRINT " ";
10110 LOCATE 1,TOPFIL:PRINT barra$;
10120 PEN 0:PAPER 1
10130 LOCATE 1,2:PRINT "
SELECCIONE CON TECLAS DE CURSOR-RETUR
N DETERMINA OPCION ";
10140 INK 1,26:PEN 1:PAPER 0
10150 GOTO 10260
10160 REM ---LECTURA DE TECLADO Y COM
PROBACION DE TECLAS
10170 TEC$="*"
10180 WHILE INSTR(valida$,TEC$)=0
10190 TEC$=INKEY$:IF TEC$="" THEN T
EC$="*"

```

```

10200 WEND
10210 REM ---MUEVE CURSOR Y ACEPTA OPCION
10220 IF TEC$=CHR$(13) THEN GOTO 10300
10230 LOCATE cod,fid:PRINT fichero$;
10240 fid=fid-(TEC$=CHR$(241) AND fid<TOPFIL)+(TEC$=CHR$(240) AND fid>4)
10250 cod=cod-15*(TEC$=CHR$(243) AND cod<60)+15*(TEC$=CHR$(242) AND cod>1)
10260 LOCATE cod,fid
10270 fichero$="":FOR n=0 TO 11:LOCATE cod+n,fid:fichero$=fichero$+COPYCHR$(#0):NEXT
10280 LOCATE cod,fid:PEN 0:PAPER 1:PRINT fichero$:PEN 1:PAPER 0
10290 GOTO 10170
10300 REM ---DEVUELVE NOMBRE FICHERO A PROGRAMA PRINCIPAL
10310 IF FID=TOPFIL THEN FICHERO$="CANCELADO"
10320 RETURN

```

```

10 REM programa demostracion
20 GOSUB 10000
30 IF fichero$(<)STRING$(8,32) THEN RUN fichero$ ELSE LOCATE 10,10:PRINT "ABORTADO"
40 END

```


Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de AMSTRAD, **MERCADO COMUN** te ofrece sus páginas para publicar los pequeños anuncios que relacionados con el ordenador y su mundo se ajusten al formato indicado a continuación.

En **MERCADO COMUN** tienen cabida, anuncios de ventas, compras, clubs de usuarios de AMSTRAD, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A.**

AMSTRAD SEMANAL.

Apartado de correos 54.062

28080 MADRID

¡ABSTENERSE PIRATAS!

Vendo los libros «Hacia la Inteligencia Artificial con **Amstrad**» y «Música y sonidos con **Amstrad**». También los siguientes juegos (versiones originales): Southern Belle; Dragonthor; Jump Jet; Beach Head; Alien 8; Exploding Fist; Decathlon; Rocky; Batalla de Midway;. Asimismo entrego cuatro cintas virgenes C-15 y vendo todo el paquete por sólo 3.000 ptas. Interesados llamar al (988) 74 41 38 y preguntar por Dani.

Vendo videojuego Trensi para TV blanco & negro y color. Dispone de tenis, hockey, frontón, squash, tiro al blanco y tiro al plato. Se entrega con los dos mandos para los jugadores, rifle desmontable, alimentador e instrucciones. Está en perfecto estado y lo vendo por tan sólo 5.000 ptas. Interesados llamar al (988) 74 41 38 y preguntar por Dani.

Vendo ordenador de bolsillo & calculadora CASIO PB 100, con unidad central, teclado, pantalla y alimentación en una sola pieza. Se incluye un amplio manual de instrucciones y un libro de programación BASIC PB 100 (ambos en castellano), así como una funda protectora y dos pilas de litio CR 2032. Está absolutamente nueva y la vendo por tan sólo 4.000 ptas. Interesados llamar al (988) 74 41 38 y preguntar por Dani.

Vendo Amstrad CPC 6128 con monitor fósforo verde. Recién comprado con garantía oficial de AMSTRAD ESPAÑA. Adjunto programas de regalo (lúdicos y de utilidades). Todo por sólo 89.900 ptas. Interesados llamar al (972) 31 53 86. Preguntar por Miquel entre las 14 y 16 horas. O bien escribir a: Miquel Gasull. Alvaro de Bazán, 2. Palamós (Gerona).

Desearía cambiar el Everyone's a Wally (original), por uno de estos títulos: Dragonthor; Exploding fist; Pole Position; Southern Belle; Stars Trike; Knight Lore; Beach Head I o Beach Head II. También me interesarían otros títulos. Gerona, 30, 2.º San Adrià de Besós (Barcelona). Tel. (93) 381 08 36. Preguntar por Fernando.

Vendo Amstrad 464 (monitor color); en perfecto estado y con garantía. Regalo 30 juegos, un joystick, y 10 libros. Precio a convenir. Escribir a: Galicia, 15 apt.601. Las Palmas de Gran Canaria.

Cambio o vendo por **Amstrad** CPC 464 fósforo verde o 50.000 ptas. por un Spectrum 48K dispuesto para su funcionamiento, con un cassette especial para ordenador más un interface tipo Kempston con salida de sonido por TV y su joystick correspondiente. Además todas las revistas de: Todospectrum, Microhobby, Micromania, Input. También un libro de Basic, uno de código máquina, y uno de juegos. Incluyo en el lote 300 juegos educativos, recreativos y culturales cuyo valor es de 500.000 ptas. y también todos los copiones existentes: Mikie, Zorro, Rambo, Comando olé toro, Paradise, Alien 8, Rocky, Exploding fist, Contabilidad doméstica, Gens 3, Mons 3, Paintbox, Tans Spres, Baudcopy, Gmnycopy, Editor, Beta Basic 3/0, Logo, etc. Llamar al tel. 375 28 81. Barcelona. Manuel Román Guerrero. Durazno, 1, 5.º puerta 4.ª Cornellà de Llobregat (Barcelona).

SERMICRO

SERVICIO TECNICO Amstrad

Barcelona
(93) 307 13 91

Sevilla
(954) 52 56 24

Valencia
(96) 154 11 43

La Coruña
(981) 22 10 12

Bilbao
(94) 444 60 87

Zaragoza
(976) 34 57 22

Málaga
(952) 32 09 44

P. de Mallorca
(971) 23 07 37

SEDE CENTRAL:

Avda. Ramón y Cajal, 107
Tel. 416 80 85

28043 MADRID

Tu tercera pieza y tu tercer número

Recorta
y pega
esta
pieza
en su lugar.



3

Recorta este número y guárdalo hasta que tengas los restantes, después pégalo en su lugar correspondiente, de forma que las sumas horizontales y verticales coincidan (15).

NUMEROS DE IMPRESION

Continuaremos hoy con la confección de algunas rutinas en código máquina, que no son posibles efectuarlas con simples llamadas al firmware.



oy le toca el turno a la impresión de números en pantalla que como todos sabemos resulta imprescindible en cualquier programa que nos propongamos hacer.

Desde Basic este tema lo podemos solucionar de una forma bien sencilla, como puede ser:

```
10 A=5000
20 PRINT A
```

o bien:

```
10 PRINT 5000
```

También nos puede interesar, imprimir dichos números en forma hexadecimal, con lo que entonces haríamos:

```
10 A=5000
20 PRINT HEX$(A)
```

o también:

```
10 PRINT HEX$(5000)
```

Pues bien, veamos cuáles son los pasos que deberemos efectuar para conseguir unas rutinas cuya función sea la impresión de números en pantallas.

Impresión decimal de números

En primer lugar describiremos la encargada de imprimir números en forma decimal.

Para ello lo primero que haremos, será posicionar el cursor en pantalla, en el lugar donde se desee imprimir el número:

```
LD HL,(POSIC)
CALL #BB75
```

donde 'POSIC' contiene la posición de pantalla donde se desea pintar.

A continuación cargamos en el registro HL el número que se desea imprimir.

```
LD HL,(NUMERO)
```

A partir de este momento empieza ya la rutina que se encarga de descifrar el número contenido en HL.

Dado que en un registro doble, el mayor número que se puede almacenar es 65535, esta rutina únicamente imprimirá números que se encuentren dentro del rango 0-65535.

Para conseguir desgranar cada uno de los dígitos de que consta el número, realizaremos la siguiente operación: Restaremos en primer lugar el valor 10000 al contenido de HL, incrementando un contador a cada resta, hasta que el contenido del registro HL sea menor que 10000. Seguidamente, efectuaremos la misma operación pero restándole 1000, después 100 y así sucesivamente.

De esta forma, si por ejemplo el número era 50000, deberemos restar el valor 10000, 5 veces, para que el contenido de HL sea menor que dicho valor, de esta forma el contador valdrá 5, con lo que habremos conseguido el primer dígito del número en cuestión.

Veamos cómo se realiza este bucle en el cual se produce la resta de 10000:

```
SCF
LD DE,10000
INC HL
LD A,47
DMIL:INC A
SBC HL,DE
JR NC,DMIL
CALL PRINT
```

en primer lugar ponemos a uno el flag Carry e incrementamos HL para que la operación 'SBC HL,DE' se



realice con normalidad, seguidamente cargamos el acumulador con 47, ya que éste es el código ASCII menos uno del valor '0', y seguidamente entramos en el bucle que decrementa HL e incrementa al contador A.

Una vez finalizado el bucle, se llama a la rutina de impresión, la cual toma el valor del acumulador y lo pinta en pantalla.

Así pues este bucle lo deberemos repetir 3 veces más, pero en este caso el contenido de DE será 1000, 100 y 10 sucesivamente.

Cuando se llegue a las unidades, no hará falta ejecutar dicho bucle, ya que únicamente deberemos sumar el contenido del acumulador con el de L, y llamar a la impresión:

```
ADD A,L
CALL PRINT
```

Como veremos a continuación en la rutina de impresión, el contenido del acumulador siempre se pone a 47, para que la impresión sea la idónea.



La rutina 'PRINT', lo primero que hace es llamar a la rutina #BB5A del firmware que es la que provoca la impresión en pantalla, a continuación carga el acumulador con 47 y resta a HL el contenido de DE, para equilibrar la última resta que se hizo:

```
PRINT:CALL #BB5A
LD A,47
JR NZ,PAS
INC HL
PAS:ADD HL,DE
INC HL
RET
```

Impresión en hexadecimal

Veamos ahora cómo funciona la rutina que imprime en pantalla números en forma hexadecimal.

En primer lugar se coloca el cursor en la posición de pantalla deseada como hemos hecho anteriormente, y a continuación se carga el registro

doble HL con el número que se desea imprimir.

Una vez hecho esto, pasamos al acumulador la parte más significativa del número es decir, la que contiene el registro H:

```
LD A,H
CALL PRHEX
```

y llamamos a la rutina que imprima dicho valor en hexadecimal.

Cuando entramos en esta rutina, en primer lugar preservamos AF, y tomamos los 4 bits más significativos del acumulador:

```
PRHEX:PUSH AF
AND #F0
```

y ahora pasamos esos bits y los desplazamos al lugar de los 4 menos significativos:

```
SRL A
SRL A
SRL A
SRL A
```

En este momento tenemos en el acumulador un número que se encuentra dentro del rango 0-15, ya

Código MAQUINA

que esos son los valores que nos pueden dar los 4 bits de menor orden.

Ahora debemos llamar a la rutina que deberá ajustar el contenido del acumulador al rango del valor ASCII de 0-F.

```
VHEX:ADD A,48
CP 58
JR C,NOSUM
```

sumamos 48 al contenido del acumulador, y si no hay Carry imprimimos ese valor, de lo contrario indicará que el número en cuestión es mayor que '9', por lo que deberemos ajustar nuevamente ese valor para que nos dé un carácter dentro del rango A-F:

```
ADD A,7
NOSUM:CALL #BB5A
RET
```

de esta forma habremos conseguido imprimir el primer dígito del número hexadecimal en cuestión.

Volvamos al lugar donde se ha llamado a la rutina 'VHEX':

```
CALL VHEX
POP AF
AND #F
VHEX: ADD A,48
```

debemos recuperar el valor del registro A que hemos almacenado previamente, y ahora tomamos los 4 bits menos significativos, para imprimir el segundo dígito de nuestro número, ya que entramos directamente en la rutina 'VHEX'.

Una vez hecho esto, se carga en el acumulador el registro L, que contenía la parte menos significativa del número que se deseaba imprimir, y se vuelve a repetir el mismo proceso descrito anteriormente, con lo cual obtendremos impresionado en pantalla el número deseado.

En las últimas líneas de cada una de las rutinas es donde hemos almacenado las variables utilizadas, que inicialmente contienen los siguientes valores:

```
POSIC:DEFW #0505
NUMERO:DEFW 5000
```

por lo tanto deberemos colocar en estas 2 variables, el lugar de impresión en pantalla y el número que se desea imprimir.


```

A000      10      ORG #A000
          20 ;
          30 ;
          40 ;IMPRIME NUMEROS DECIMALES
          50 ;
          60 ;

```

```

A000 2A49A0      70      LD HL,(POSIC)
A003 CD75BB      80      CALL #BB75
A006 2A4BA0      90      LD HL,(NUMERO)
A009 37          100     SCF
A00A 111027     110     LD DE,10000
A00D 23          120     INC HL
A00E 3E2F       130     LD A,47
A010 3C          140     DMIL: INC A
A011 ED52       150     SBC HL,DE
A013 30FB       160     JR NC,DMIL
A015 CD3EA0     170     CALL PRINT
A018 11E803     180     LD DE,1000
A01B 3C          190     MIL: INC A
A01C ED52       200     SBC HL,DE
A01E 30FB       210     JR NC,MIL
A020 CD3EA0     220     CALL PRINT
A023 116400     230     LD DE,100
A026 3C          240     CIEN: INC A
A027 ED52       250     SBC HL,DE
A029 30FB       260     JR NC,CIEN
A02B CD3EA0     270     CALL PRINT
A02E 110A00     280     LD DE,10
A031 3C          290     DIEZ: INC A
A032 ED52       300     SBC HL,DE
A034 30FB       310     JR NC,DIEZ
A036 CD3EA0     320     CALL PRINT
A039 85          330     ADD A,L
A03A CD3EA0     340     CALL PRINT
A03D C9          350     RET
A03E CD5ABB     360     PRINT: CALL #BB5A
A041 3E2F       370     LD A,47
A043 2001       380     JR NZ,PAS
A045 23          390     INC HL
A046 19          400     PAS: ADD HL,DE
A047 23          410     INC HL
A048 C9          420     RET
A049 0505       430     POSIC: DEFW #0505
A04B 0F27       440     NUMERO: DEFW 9999

```

```

CIEN  A026  DIEZ  A031  DMIL  A010
MIL   A01B  NUMERO A04B  PAS   A046
POSIC A049  PRINT A03E

```

```

10 REM *PROGRAMA CARGADOR*
20 FOR N=&A000 TO &A04D
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>&1ED5 THEN PRINT "ERROR
  EN DATAS"
70 DATA 42,73,160,205,117,187,42
80 DATA 75,160,55,17,16,39,35
90 DATA 62,47,60,237,82,48,251
100 DATA 205,62,160,17,232,3,60
110 DATA 237,82,48,251,205,62,160
120 DATA 17,100,0,60,237,82,48
130 DATA 251,205,62,160,17,10,0
140 DATA 60,237,82,48,251,205,62
150 DATA 160,133,205,62,160,201,205
160 DATA 90,187,62,47,32,1,35
170 DATA 25,35,201,5,5,15,39,0

```

```

A000      10      ORG #A000
          20 ;
          30 ;
          40 ;IMPRIME NUMERO HEXADECIMALES
          50 ;
          60 ;

```

```

A000 2A2FA0      70      LD HL,(POSIC)
A003 CD75BB      80      CALL #BB75
A006 2A31A0      90      LD HL,(NUMERO)
A009 7C          100     LD A,H
A00A CD12A0     110     CALL PRHEX
A00D 7D          120     LD A,L
A00E CD12A0     130     CALL PRHEX
A011 C9          140     RET
A012 F5          150     PRHEX: PUSH AF
A013 E6F0       160     AND #F0
A015 CB3F       170     SRL A
A017 CB3F       180     SRL A
A019 CB3F       190     SRL A
A01B CB3F       200     SRL A
A01D CD23A0     210     CALL VHEX
A020 F1          220     POP AF
A021 E60F       230     AND #F
A023 C630       240     VHEX: ADD A,48
A025 FE3A       250     CP 58
A027 3802       260     JR C,NOSUM
A029 C607       270     ADD A,7
A02B CD5ABB     280     NOSUM: CALL #BB5A
A02E C9          290     RET
A02F 0505       300     POSIC: DEFW #0505
A031 CCFA       310     NUMERO: DEFW #FACC

```

```

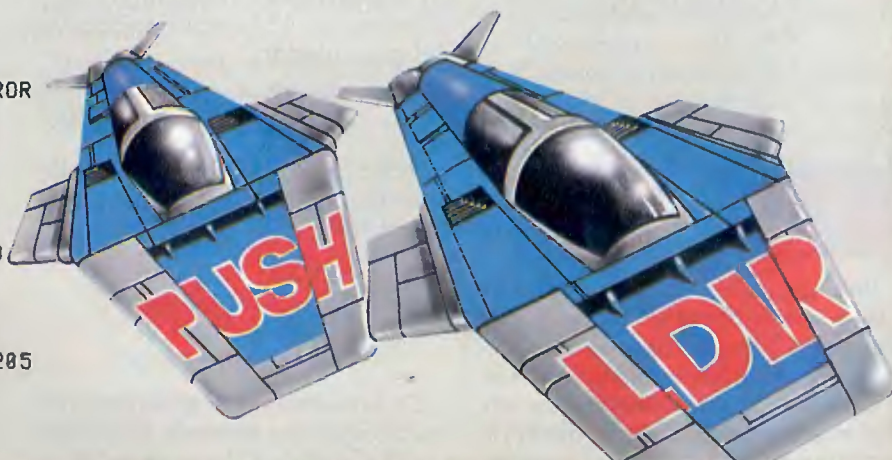
NOSUM A02B  NUMERO A031  POSIC  A02F
PRHEX  A012  VHEX   A023

```

```

10 FOR N=&A000 TO &A033
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<>&1ACA THEN PRINT "ERROR
  EN DATAS"
60 DATA 42,47,160,205,117,187,42
70 DATA 49,160,124,205,18,160,125
80 DATA 205,18,160,201,245,230,240
90 DATA 203,63,203,63,203,63,203
100 DATA 63,205,35,160,241,230,15
110 DATA 198,48,254,58,56,2,198
120 DATA 7,205,90,187,201,5,5
130 DATA 204,250,0,0,0,0,0

```



LUGAR
DESTINADO
PARA
EL SELLO

HOBBY PRESS, S. A.

Apartado de Correos N.º 232

ALCOBENDAS

(Madrid)

d, los
o de
de
sobre

undo
el
aje
más



Si no lo encontraras en tu kiosco, puedes solicitarlo directamente a nuestra editorial, llamando por teléfono (91) 734 65 00, o bien, rellenando el cupón.

Recorta o copia este cupón y envíalo a Hobby Press, Apartado de Correos 232, Alcobendas (Madrid)

SI deseo recibir en mi domicilio el especial de Amstrad n.º 3 al precio de 350 ptas.

APELLIDOS
NOMBRE
DOMICILIO
LOCALIDAD
TELÉFONO

¿ERES SUScriptor DE MICROHOBBY AMSTRAD?

Forma de pago:

☐ Mediante talón bancario adjunto a nombre de Hobby Press, S. A.

☐ Mediante giro postal n.º

☐ Mediante tarjeta de crédito n.º

☐ Visa ☐ Master Charge ☐ American Express

Fecha de caducidad de la tarjeta

☐ Contra reembolso del envío (suponen 100 ptas. más de gastos de envío)

FECHA DE NACIMIENTO

PROVINCIA

CODIGO POSTAL

¿ERES SUScriptor DE MICROHOBBY AMSTRAD?

Forma de pago:

☐ Mediante talón bancario adjunto a nombre de Hobby Press, S. A.

☐ Mediante giro postal n.º

☐ Mediante tarjeta de crédito n.º

☐ Visa ☐ Master Charge ☐ American Express

Fecha de caducidad de la tarjeta

☐ Contra reembolso del envío (suponen 100 ptas. más de gastos de envío)

Fecha y firma

Solicitud de cintas y n.ºs atrasados

A000

■ Deseo recibir en mi domicilio los siguientes números atrasados de Microhobby Amstrad al precio de 160 ptas. cada uno _____
El número 4 está agotado.

■ Deseo recibir la(s) cinta(s) de programas que indico a continuación. Cada una de estas cintas lleva grabados los programas publicados por Microhobby Amstrad durante cuatro números consecutivos (1 al 4, 5 al 8, 9 al 12, etc.) y su precio es de 756 ptas. (IVA incluido).

Números del _____ al _____ (Inclusive) Números del _____ al _____ (Inclusive)
Números del _____ al _____ (Inclusive) Números del _____ al _____ (Inclusive)
Números del _____ al _____ (Inclusive) Números del _____ al _____ (Inclusive)
NOMBRE _____ Fecha de nacimiento _____
APELLIDOS _____ Domicilio _____

LOCALIDAD _____ PROVINCIA _____
C. POSTAL _____ TELÉFONO _____ PROFESIÓN _____

¿Eres suscriptor de Microhobby Amstrad?

Marco con una (x) en el casillero correspondiente la forma de pago que más me conviene.

☐ Tarjeta de crédito n.º _____

☐ Visa ☐ American Express ☐ Master Charge

Fecha y firma _____

Fecha de caducidad de la tarjeta _____

☐ Talón bancario a nombre de HOBBY PRESS, S. A.

☐ Giro Postal a nombre de HOBBY PRESS, S. A., N.º del giro _____

☐ Contra reembolso (sólo para números atrasados, supone 75 pesetas de gastos de envío).

A000 2A49A0
A003 CD75BB
A006 2A4BA0
A009 37
A00A 111027
A00D 23
A00E 3E2F
A010 3C
A011 ED52
A013 30FB
A015 CD3EA0
A018 11E803
A01B 3C
A01C ED52
A01E 30FB
A020 CD3EA0
A023 116400
A026 3C
A027 ED52
A029 30FB
A02B CD3EA0
A02E 110A00
A031 3C
A032 ED52
A034 30FB
A036 CD3EA0
A039 85
A03A CD3EA0
A03D C9
A03E CD5ABB
A041 3E2F
A043 2001
A045 23
A046 19
A047 23
A048 C9
A049 0505
A04B 0F27

270 CALL PRINT
280 LD DE,10
290 DIEZ: INC A
300 SBC HL,DE
310 JR NC,DIEZ
320 CALL PRINT
330 ADD A,L
340 CALL PRINT
350 RET
360 PRINT: CALL #BB5A
370 LD A,47
380 JR NZ,PAS
390 INC HL
400 PAS: ADD HL,DE
410 INC HL
420 RET
430 POSIC: DEFW #0505
440 NUMERO: DEFW 9999

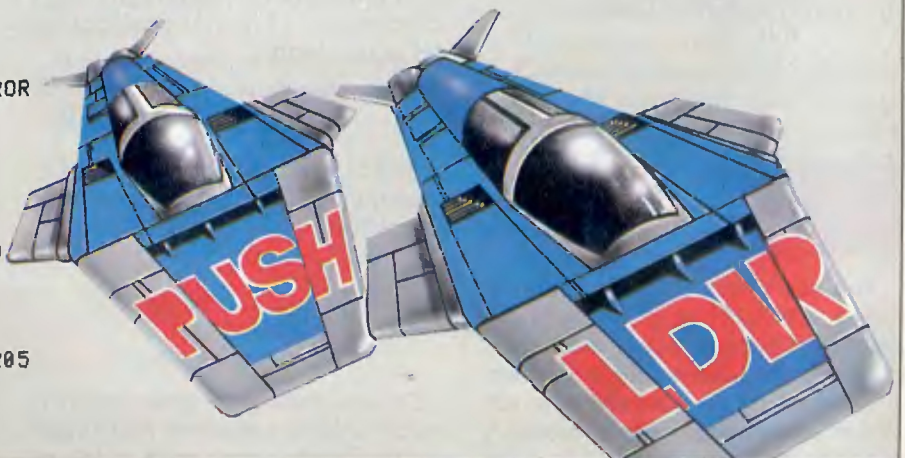
A029 C607 270 ADD A,7
A02B CD5ABB 280 NOSUM: CALL #BB5A
A02E C9 290 RET
A02F 0505 300 POSIC: DEFW #0505
A031 CCFA 310 NUMERO: DEFW #FACC

NOSUM A02B NUMERO A031 POSIC A02F
PRHEX A012 VHEX A023

CIEN A026 DIEZ A031 DMIL A010
MIL A01B NUMERO A04B PAS A046
POSIC A049 PRINT A03E

10 FOR N=&A000 TO &A033
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<>&1ACA THEN PRINT "ERROR
EN DATAS"
60 DATA 42,47,160,205,117,187,42
70 DATA 49,160,124,205,18,160,125
80 DATA 205,18,160,201,245,230,240
90 DATA 203,63,203,63,203,63,203
100 DATA 63,205,35,160,241,230,15
110 DATA 198,48,254,58,56,2,198
120 DATA 7,205,90,187,201,5,5
130 DATA 204,250,0,0,0,0,0

10 REM *PROGRAMA CARGADOR*
20 FOR N=&A000 TO &A04D
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>&1ED5 THEN PRINT "ERROR
EN DATAS"
70 DATA 42,73,160,205,117,187,42
80 DATA 75,160,55,17,16,39,35
90 DATA 62,47,60,237,82,48,251
100 DATA 205,62,160,17,232,3,60
110 DATA 237,82,48,251,205,62,160
120 DATA 17,100,0,60,237,82,48
130 DATA 251,205,62,160,17,10,0
140 DATA 60,237,82,48,251,205,62
150 DATA 160,133,205,62,160,201,205
160 DATA 90,187,62,47,32,1,35
170 DATA 25,35,201,5,5,15,39,0



Recorte o copie este cupón

AMSTRAD ESPECIAL N.º 3
DESDE HOY EN TU
KIOSCO

EL FUTURO EMPEZO AYER

Descubre el fascinante mundo de los robots: Su utilidad, los últimos avances y cómo afectarán a tu vida. ● El juego de la vida: nacimiento, evolución y muerte en un mundo de fantasía. ● Simuladores de vuelo: Un trepidante vuelo sobre todos los simuladores de combate para AMSTRAD. ● HISOFT-C: El lenguaje de alto nivel más rápido del mundo para tu CPC. ● El mundo de bloques: Compresión del lenguaje natural desde Basic. ● Forth: Todo un lenguaje especialmente diseñado para ti. ● Todo esto y mucho más en AMSTRAD ESPECIAL NUMERO TRES.



Si no lo encontraras en tu kiosco, puedes solicitarlo directamente a nuestra editorial, llamando por teléfono (91) 734 65 00, o bien, rellenando el cupón.

Recorta o copia este cupón y envíalo a Hobby Press, Apartado de Correos 232, Alcobendas (Madrid)

SI deseo recibir en mi domicilio el especial de Amstrad n.º 3 al precio de 350 ptas.

APELLIDOS
NOMBRE
DOMICILIO
LOCALIDAD
TELÉFONO

¿ERES SUScriptor DE MICROHOBBY?

Forma de pago:
☐ Mediante talón bancario a junto a nombre de Hobby Press, S. A.
☐ Mediante giro postal n.º
☐ Visa ☐ Master Charge ☐ American Express
Fecha de caducidad de la tarjeta
☐ Contra reembolso del envío (suponen 100 ptas. más de gastos de envío)

FECHA DE NACIMIENTO
PROVINCIA
CODIGO POSTAL
CODIGO AMSTRAD?

Fecha y firma

El único ordenador
concebido para sustituir
a la máquina de escribir.



AMSTRAD PCW 8256

UN COMPLETO EQUIPO QUE INCLUYE:

- Unidad Central (256 K RAM) • Teclado en castellano
- Unidad de disco (180 K por cara) • Pantalla de alta resolución • Impresora alta calidad (NLQ)
- Programas: • Procesador de textos, sistema Operativo CP/M Plus, Mallard Basic con JET SAM para ficheros indexados, lenguaje DR LOGO.

PROGRAMAS PROFESIONALES

- Contabilidades • Almacenes • Facturación • HOJAS DE CALCULO: **Multiplán**, Supercalc 2, Cracker, Plannercalc. BASES DE DATOS: **DBase II**, Amsfile, Flexifile, **Boriar**. LENGUAJES: Cobol, Fortran, Pascal MT +, Pilot, etc.

También disponible la versión PCW 8512, con 512 K RAM y 2º disco de 1 MBYTE incorporado.

P.V.P. 149.900,— Ptas. + IVA

SOLICITE DEMOSTRACION EN:

División informática de **El Corte Inglés**, División **Online** de GALERIAS, Tiendas especializadas en informática y Equipos de oficina.

NOTA: El Amstrad también puede ser utilizado como "Terminal Inteligente" de grandes equipos informáticos.

¡¡ Increíble !!

AMSTRAD

ESPAÑA

GRUPO INDESCOMP

C/. Aravaca, 22. 28040 Madrid. Tel. 459 30 01. Télex 47660 INSC E. Fax 459 22 92. Delegación en Cataluña: C/. Tarragona, 110. Tel. 325 10 58. 08015 Barcelona.